Contract Deliverable for Task 24

Report on the System Requirements for the Integrated Database for Mental Health and Substance Abuse Treatment Service Project

Under CSAT/CMHS
Contract # 270-96-007
Project Officer: Joan Dilonardo

August 29, 1997

TABLE OF CONTENTS

LIST OF FIGURES

Executive Summary

This report addresses the system requirements for building fully integrated databases of mental health, alcohol/other drug (MH/AOD), and Medicaid data for three participating states under CSAT/CMHS contract (270-96-007). An integrated database, containing cost and usage information on all medical services for recipients of MH/AOD services within that state, will be constructed for each of the three participating states: Delaware, Oklahoma, and Washington. This report deals with the processing design, or detailed architecture, proposed to build these databases. In developing the design, a great deal of consideration has been placed on creating databases that retain most of the information available from each state while also creating databases that are fairly uniform across the three states.

Drawing upon previous deliverables, we have developed the detailed architecture - the processing design - required to build the three databases. We present the initial design describing the flow of data and highlighting steps important in creating the integrated databases, including a brief examination of an HTML based documentation system. This design addresses the second significant issue identified in the Task 22 report: linking client records from disparate sources. This is a difficult process because there generally are no universal client identifiers available to control the linking process. To compensate for the lack of a universal identifier, we have designed linking methods which are quite complex. We explain the linking conceptually, and then detail actual process. Beyond the issue of linking client records, this report identifies the additional database building issues of associating costs with MH/AOD services and unduplicating services and usage counts between MH/AOD data and Medicaid data. Quality assurance measures to detect and correct errors during the database building process are also described. Finally, we discuss resources needed by the project, in both computer hardware and programming time.

The initial processing plan is a vital step towards building the databases. It provides the specifications, or blueprint, for transforming numerous data files from multiple agencies into the integrated, MH/AOD databases. The design represents our current knowledge of the data available from Delaware, Oklahoma, and Washington and the issues required with client linking. However, modifications to this design are likely. While minor modifications to the design are likely, any changes will not alter the design's general structure. The final design will reflect the initial design proposed here, as well as extensive testing and revisions optimized for the actual data.

Introduction

This report details the system requirements and proposed processing design, for the integrated database of MH/AOD services. The CSAT/CMHS contract (270-96-007) calls for integrated databases to be constructed for each of three participating states: Delaware, Oklahoma, and Washington. Each database will contain cost and usage information on all medical services received by any beneficiary of MH/AOD services within that state, and will be available to SAMHSA/CSAT/CMHS and state agency staff for summarization and analysis of MH/AOD services. This report deals with the processing design, or detailed architecture, proposed to build these databases. The report first presents an overview of the design, offering a high-level view and describing in general terms, the necessary processes required to construct the databases. Expected issues and difficulties are considered, as well as a brief examination of an HTML based documentation system. Quality assurance measures to detect and correct errors during the database building process are also addressed. Following the overview the particulars of each processing stage are described, program by program. Discussions of the programs include the flow of data and address crucial operations. A thorough examination of the client linking process, including the scoring process, is presented. Finally, we discuss resources needed by the project, in both computer hardware and programming time.

Prior contract deliverables have:

- detailed types of data collected by, and available from the three states (Task 21),
- presented alternative processing structures, recommending a hybrid SAS/relational architecture (Task 22), and
- discussed the architectural requirements (Task 23).

Determining the logical structure of the final databases was one of the significant issues identified in the Task 22 report. In analyzing the necessary steps for processing data and constructing these databases, the deliverable described two alternative system architectures: an open Relational Database Management System (RDBMS) and the Statistical Analysis System (SAS). Because of the data processing capabilities inherent in SAS, and the complexity of data manipulations and transformations required by the project, SAS was deemed the superior choice. Task 22 recommended a hybrid architecture using SAS to process and build the final database, but also making the data available in a normalized form for use in a RDBMS. Subsequent discussions with representatives from the three participating states confirmed the appropriateness of this view, and SAS was selected as the architecture for processing the data. The deliverable for Task 23 discussed feedback concerning the recommended architecture, which was favorable, and the system implications of that architecture.

The remainder of this report addresses the system requirements and proposed processing design, for the integrated database. The conception of this design has truly been a group effort by the MEDSTAT team. A considerable effort has been made to provide a well thought out and detailed processing plan. But the reader should bear in mind that this is a starting point only: the design is not “carved in stone.” The nature of the project is such that it requires a flexible system. The goal of this report is to define and describe the system requirements from a high level perspective, realizing that aspects of the requirements will change as the database is further developed. A great deal of testing will take place with the data, and revisions to accommodate operational data are likely. Thanks are due to the analysts with the Washington Department of Social and Health Services, Division of Research and Analysis for sharing their experiences in linking client records. We found many similarities to their work in building integrated databases, particularly their Client Registry and First Steps databases. The Washington work confirmed our strategy and highlighted difficulties we are likely to face.

Processing Overview

This section provides a general, high level overview of the proposed processing plan. In a later section we will provide a more detailed view, examining the steps we propose for creating the integrated database. Our goal, once again, is to create three separate integrated databases of MH/AOD and Medicaid data for clients receiving MH/AOD services. A database will be created for each of three states: Washington, Oklahoma, and Delaware. We have designed the operation of building each database as an iterative process. Each new database build will make use of linkage information ascertained during the previous database build. The iterative design will allow us to process additional years of data while maintaining previously determined links, should the need arise.

The complex data manipulations and transformations required with this project have lead us to the choice of SAS for processing data and building the three integrated databases. In creating these databases, there are three critical operations:

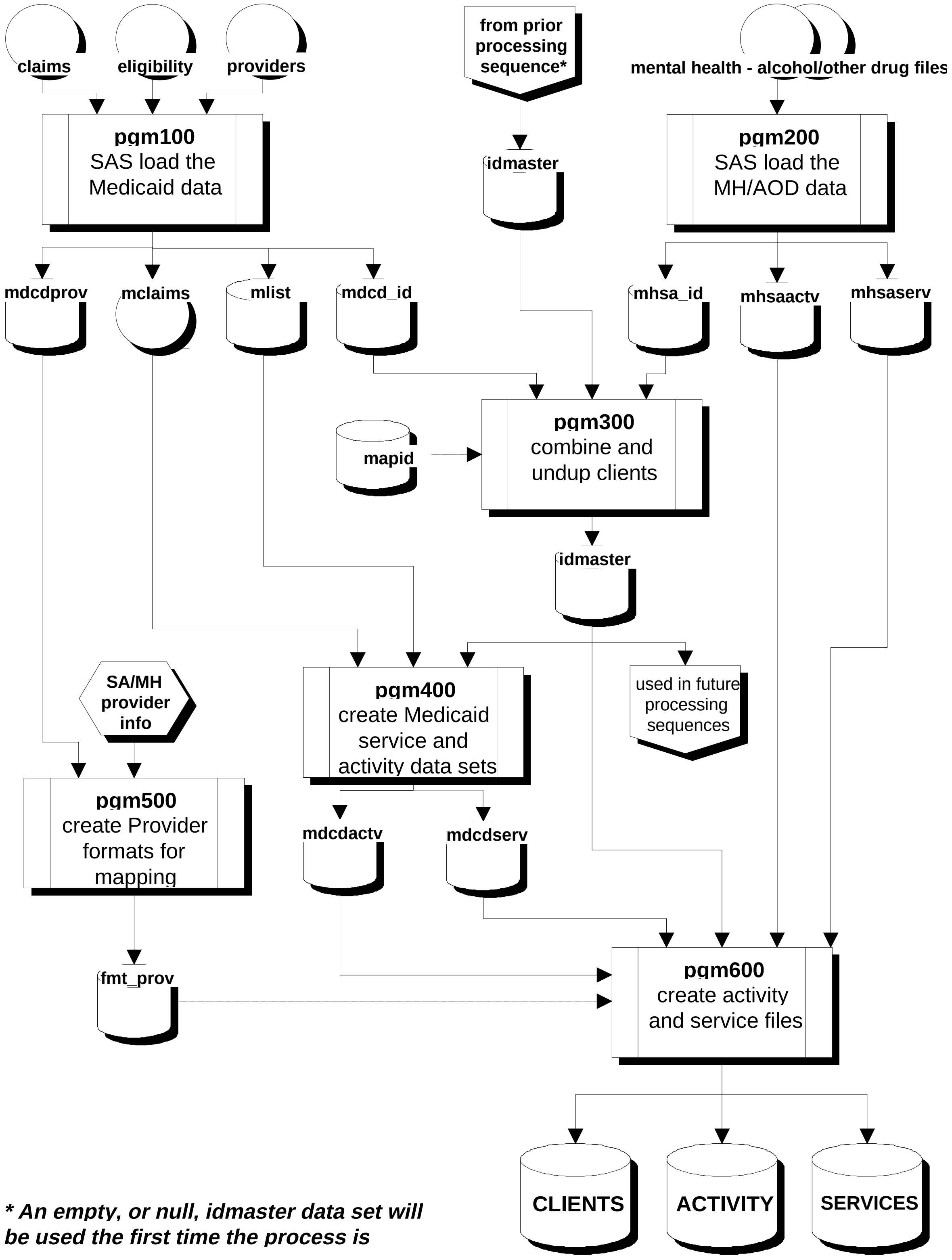
- Associating costs with MH/AOD services,
- Linking client records, and
- Unduplicating Medicaid and MH/AOD services.

These operations will be performed in programs 200, 300, and 600 respectively. A graphical representation of the process is provided in Figure 1 on page 6. Each program will address one general process, such as SAS loading Medicaid data or linking clients. In turn, each program will be divided into two or more steps, each dealing with specific tasks within the general process of the program such as SAS loading the Medicaid *Eligibility* data. Technically, each “step” will be a separate SAS program, while the “program” will be a UNIX shell script, such as a perl[1] script, connecting the SAS “steps”.

To the extent possible, programs will be standardized between the three states. Because of the variety of data sources, less uniformity will be possible during the early processing stages (where data is loaded into SAS data sets) than in the later programs. Yet a uniform approach is possible even the early programs. For example, there is a common overall structure shared by all the states for their Medicaid data. While

[1] perl (Practical Extraction and Report Language) - is a general purpose, interpreted scripting language available on UNIX systems, useful for executing system calls and other commands.

Figure 1 - Integrating Medicaid and MH/AOD Data


claims
eligibility
providers
from prior processing sequence*
mental health - alcohol/other drug files
pgm100 SAS load the Medicaid data
idmaster
pgm200 SAS load the MH/AOD data
mdcdprov
mclaims
mlist
mdcd_id
mhsa_id
mhsaactv
mhsaserv
mapid
pgm300 combine and undup clients
idmaster
SA/MH provider info
pgm400 create Medicaid service and activity data sets
used in future processing sequences
pgm500 create Provider formats for mapping
mdcdactv
mdcdserv
fmt_prov
pgm600 create activity and service files
CLIENTS
ACTIVITY
SERVICES

***** An empty, or null, idmaster data set will be used the first time the process is***

Medicaid data from all three states vary in layout and specific content, each state provides Medicaid data in the form of eligibility, claims, encounter, and provider files. That common structure allows us to set up a general framework for loading Medicaid data. Benefits of a standardized framework include more efficient program development and the ability to provide a set of programs usable by states not participating in this project.

When complete, the integrated database will contain information for all medical services received by any recipient of MH/AOD services. Both the Medicaid and state MH/AOD data are needed for identifying this population. While many clients in the final database will have received services from a state MH/AOD agency, other clients will have used Medicaid services directly, without seeking assistance from another state agency. Identifying the former group is straightforward using the client list supplied by the MH/AOD agencies. Identifying the later group is more problematic and will require two passes through the Medicaid claims data, as discussed below in the section titled “Medicaid Data”.

Quality Assurance

Quality analysis will be performed throughout the processing sequence. Because SAS, which includes a wide variety of data analysis and presentation functions, will be used for processing the data, all SAS analysis and reporting functions will be available while processing the data. This will allow us to integrate statistical and report procedures within the programs for a continuous and ongoing quality assurance process. An ongoing process will greatly help with the detection and correction of errors. Reviews and data checks will be performed after executing each of the six separate programs to ensure data integrity. Much of the quality assurance efforts will focus on detecting problems with the incoming data and catching programming errors from operations such as cross walks. If errors are detected, corrections will be made before moving on to the next program. In addition, files from intermediate steps will be retained until the final integrated database has passed internal reviews and been approved. This will enable us to restart the processing at any point within the processing stream to correct errors.

Medicaid Data

Medicaid data presents the most voluminous files for this project, particularly the eligibility and claim files which will provide service level information. The provider file will be used for mapping services types, as well as unduplicating services between Medicaid and MH/AOD sources. And the eligibility file will be used for client information including identifiers and other socio-demographic data. The first program in the processing sequence, program 100, will read the Medicaid data into SAS data sets. This is referred to as SAS loading the data. During the SAS load of the eligibility data, the data is "flattened." For a single recipient, records for multiple eligibility periods will be combined, creating an array of eligibility start and end dates, on a single observation (the SAS equivalent of a record). Output from the eligibility file will include identification and demographic information, and will be sorted by the Medicaid ID (*MCAID_ID*). This data will pass to program 300 for linking and Unduplicating client records.

Program 100 will also load claims and provider information into SAS data sets. All Medicaid claim records will be loaded and saved for use later in the process during program 400. While loading the claims, recipients of MH/AOD services will be identified based on pre-determined diagnoses codes, procedure codes, and other relevant criteria such as type of service and provider type. Appendix A contains a condensed, preliminary list of these criteria. In program 400, this client list will be combined with a client data set from the linking process to collect claims for the final database. Program 400 will collect claims using the data set of Medicaid claims loaded in program 100. From the extracted claims, program 400 will then create data sets of Medicaid service level data and summarized activity data. This data will proceed into program 600 for use in creating the final, integrated database.

Provider data, loaded in program 100, will be used in program 500 to create SAS formats for mapping between Medicaid and MH/AOD provider numbers. These formats will be used in program 600 in creating the integrated database.

State MH/AOD Data

The least amount of program standardization will be possible with the loading of the MH/AOD data. Since each of the three states involved with the project has adopted their own approach to best serve their state's citizenry, the programs for processing MH/AOD data require more customization than the programs for processing Medicaid data. For example, Washington state uses two separate agencies to provide mental health services and alcohol/other drug services, each of which use client identifiers unique to that agency. Oklahoma has one agency providing both mental health and alcohol/other drug services. Delaware employs one agency to provide mental health and alcohol/other drug services to adults, and another agency to provide those services to children and adolescents. Because the data from each participating state is uniquely structured, it will be necessary to create separate programs for each state's data. For each state, this will be program 200.

Program 200 will load MH/AOD data into SAS data sets. In general, the program will read the raw data files and create three SAS data sets of

- service level,
- summarized activity, and
- ID/demographics data.

The ID data set will contain one observation for each state ID[2] (*STATE_ID*) found on the incoming files. This data will be used in program 300 for linking and unduplicating[3] client records. MH/AOD services will be summarized in the activity data set. Service level data, where available, will provide more detailed information for each MH/AOD service. Any and all service level data provided by the states will be retained. Unfortunately, service level data is not always available because of data limitations with some participating states. Activity and service level data will move on to program 600 which creates the final, integrated database.

We anticipate that direct cost information will not be available for many MH/AOD services. When an actual dollar cost is included with the state's data, we will of course use that dollar amount. However, when cost details are missing it will be necessary to estimate costs. This will take place while processing the MH/AOD data during program 200. for a number of services. We will work with each state agency to determine the best way to associate costs. One potential method would use the unit count from the data and an average cost per unit estimate derived from the agency budgets. Budget information may come in a variety of forms, and vary by state and by agency. Some budgets will allocate dollars for specific services, while other agencies may receive block funding. We will incorporate budget information at the most detailed level possible. Overhead costs, such as agency headquarters and community education costs, will be excluded from cost estimates where possible.

Linking and Unduplicating Clients

The third program in the sequence, program 300, will integrate the MH/AOD and Medicaid identification/demographics data, link and unduplicate clients, and assign client IDs. Medicaid identification data flows from program 100, while the MH/AOD

[2] The ID variable assigned by the various state MH/AOD agencies will be loaded as *STATE_ID*.

[3] Linking and unduplicating will be described with the program 300 description.

identification data continues on from program 200. The methodology of linking and unduplicating clients is both complex and essential to creating the integrated database. The process will be explained in greater depth in the “Processing Details” section. The general overview includes:

- unduplicating the Medicaid client data,
- linking the Medicaid client data with the MH/AOD client data, and
- unduplicating MH/AOD client data not linking with any Medicaid client data

The unduplicating and linking processes are similar: both involve matching and comparing a client observation with observations for many other clients. But where unduplicating involves a single data set, linking uses two data sets. In both cases, clients are matched and those matches are scored as to the quality of the match. The higher the score, the greater the certainty that the two client observations represent one actual client. Scoring will be based upon a variety of client identification and demographic information, and will be discussed further in the section “Processing Details”.

Program 300 will create a “master” ID data set containing the client ID, newly assigned within the program, and demographic information for the client. This master file will also contain the source IDs from the Medicaid and MH/AOD files associated with each client ID. This mapping will be needed in program 600 to combine the Medicaid and MH/AOD activity and services data and build the final, integrated database. The master ID data set will also retain ID mappings from one processing stream to the next, serving as the connection between iterations of the processing stream.

Creating the Final Database

The final database, consisting of the **`CLIENTS`**, **`ACTIVITY`**, and **`SERVICES`** SAS data sets will be created in program 600 by combining the Medicaid and MH/AOD activity and services data, from programs 200 and 400 with the master ID data from program 300. These data sets will be described shortly. Services and activities aggregated to *`STATE_ID`* and *`MCAID_ID`* in the earlier programs will be combined and aggregated to client IDs (*`CLIENT_ID`*), as mapped in program 300. Services counted in both the Medicaid and the MH/AOD data will be unduplicated. Unduplication will be based on types of service, dates of service, and providers (using formats created in program 500). A final program will create the normalized text data discussed in the Task 22 report. Program 700, a “postproduction” program, will convert the wide SAS data sets into third normal form[4], raw data files. The resulting series of text files will be suitable for importing into a variety of relational database packages using a bulk copy utility.

Our recommendation is that the final, integrated database will contain at least three SAS data sets, linkable through a common ID (*CLIENT_ID*):

- **CLIENTS** - containing *CLIENT_ID* along with socio-demographic information for the client. The **CLIENTS** data set will contain one observation for each *CLIENT_ID*.

- **ACTIVITY** - containing *CLIENT_ID* with summarized charges and counts of services. The **ACTIVITY** data set will contain multiple observations for each *CLIENT_ID*, potentially based upon the month of the activity.

- **SERVICES** - containing *CLIENT_ID* with diagnosis and procedure codes, charges, and unit information for individual services. This may ultimately be several separate data sets if the data elements are too diverse to construct one cohesive data set. Possible divisions for separate data sets may be mental health, alcohol and other drugs, inpatient, prescription drugs, long term care, or other medical services. The **SERVICES** data set(s) will contain multiple observations for each *CLIENT_ID*, differentiated by service dates and providers.

Figure 2 describes the relationships between the data sets.

Documentation

Documentation for the integrated database will include descriptive contents of each data set as well as a complete data dictionary. The data dictionary will provide a complete variable description. Included in the data dictionary will be:

- a depiction of the coding schemes,

- frequencies for discrete variables and ranges for continuous variables, and

- a list of SAS formats constructed for use with the variable.

[4] A relational database is said to be in normal form if it satisfies certain constraints. E.F. Codd's original work defined three such forms.

Because of other project activity aimed at publishing project developments on the SAMHSA World Wide Web site, we are leaning towards creating the documentation system as a series of linked HTML pages in an "Intranet" setting. HTML pages will allow users to create and follow their own individual information threads to receive documentation pertinent to their uses. The resulting system would be powerful and flexible, yet available to all users regardless of their hardware or operating system. The documentation files and directory structure can be copied to a CD for delivery to SAMHSA, and with little labor involved incorporated onto the SAMHSA web server. Subsets of the documentation may be individualized for delivery to the participating states. Figure 3 on page 13 is an example of a data dictionary page.

Figure 2 - Data Set Relationships in the Integrated Database

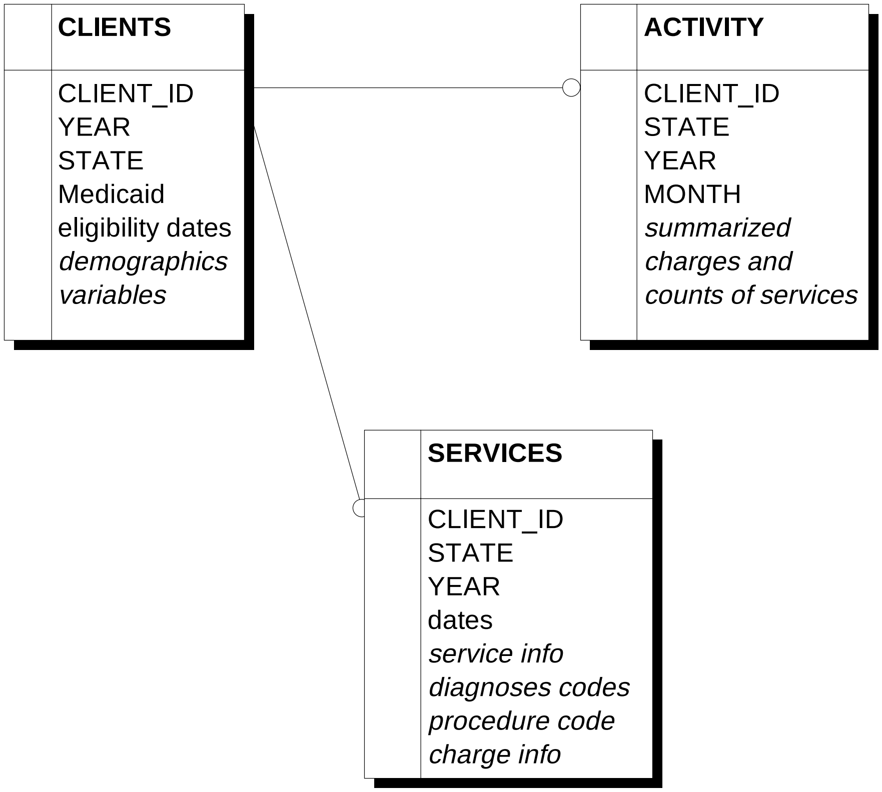
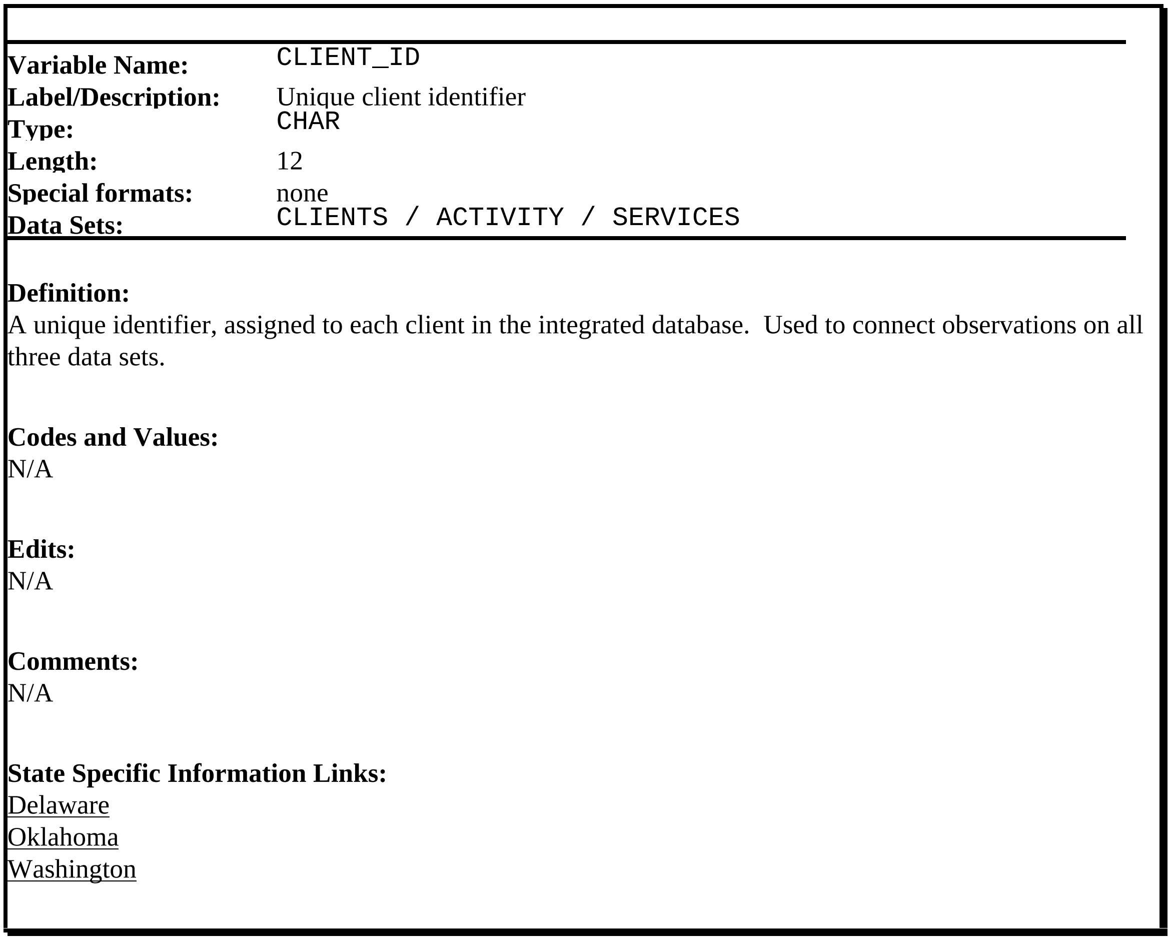


Figure 3 - Example of a Data Dictionary Page



Variable Name:	CLIENT_ID
Label/Description:	Unique client identifier
Type:	CHAR
Length:	12
Special formats:	none
Data Sets:	CLIENTS / ACTIVITY / SERVICES

Definition:
A unique identifier, assigned to each client in the integrated database. Used to connect observations on all three data sets.

Codes and Values:
N/A

Edits:
N/A

Comments:
N/A

State Specific Information Links:
Delaware
Oklahoma
Washington

Processing Details

In the following sections, we will provide a detailed view of the proposed processing plan, elaborating the details and issues of each processing step. These sections will examine the steps required to create the integrated database. Once again, our goal is to create three separate integrated databases of Medicaid and non-Medicaid data for clients receiving MH/AOD services in each of the three participating states: Washington, Oklahoma, and Delaware.

Program 100

The purpose of program 100 is to load state Medicaid data into SAS data sets. Data created will be used in later programs to link and unduplicate clients, and to build the integrated databases. As inputs, the program will expect Medicaid Eligibility, Claims, Encounters, and Provider files. The program will also use file lists of diagnosis and procedure codes to identify recipients of MH/AOD services. These files will be compiled by MEDSTAT using the coding schemes normally used by each particular state, including ICD-9-CM, HCPCS, CPT-4, and state-specific codes). Data sets output from the program will be:

- a listing of all Medicaid eligibles (clients) - **MDCD_ID** - used in program 300,
- a listing of Medicaid clients who have received any MH/AOD services - **MLIST** - used in program 400,
- all Medicaid claims/encounters - **MCLAIMS** - used in program 400, and
- a listing of Medicaid provider information - **MDCDPROV** - used in program 500.

The data sets **MDCD_ID**, **MLIST**, and **MCLAIMS** will be sorted by the Medicaid ID variable: *MCAID_ID*. The Medicaid provider data set - **MDCDPROV** - will be sorted by *MDCDPROV*, the Medicaid provider ID variable.

Processing Steps

Program 100 is the first program in the processing sequence, and will be divided into three steps which will load the claims/encounter data, the eligibility data, and the provider data. Figure 4, beginning on page 16, demonstrates the flow of data through the program. The first step (110) will load the Medicaid claims and encounter data, creating data sets with all Medicaid claims and listing all Medicaid clients with at least one MH/AOD claim. Before reading the claims, SAS formats will be created from the diagnosis and procedure codes, and other relevant information (e.g. type of service) to assist in identifying claims and clients with MH/AOD related services. See Appendix A for a condensed list of these criteria. These formats will be used to identify recipients of MH/AOD services and build the list of those clients. This step will output two data sets:

- **MCLAIMS** - all Medicaid claims, and
- **MLIST** - clients with MH/AOD claims.

Both data sets will be sorted by the variable *MCAID_ID*. The claims data will also be sorted, within Medicaid ID, by claim date and provider ID.

The next two steps (120 and 130) will load the eligibility and provider data. Eligibility data will be used to build the list of Medicaid clients - the data set **MDCD_ID**. This data set will contain identifying variables and other demographic information. To prepared for linking and unduplicating clients, identifying variables will be cleaned and standardized. Cleaning and standardizing operations will include:

- capitalizing name variables and removing “extraneous” characters such as spaces and apostrophes, and
- converting categorical variables such as *GENDER* to a common coding scheme (e.g. “F” for female and “M” for male).

Multiple records for multiple eligibility periods will be condensed into one observation containing arrays of eligibility dates and other information that can change across eligibility periods, such as eligibility group and capitation status. The **MDCD_ID** data set will be sorted by Medicaid ID variable (*MCAID_ID*). The Medicaid provider data set - **MDCDPROV** - will be sorted by Medicaid provider ID.

Figure 4 - Program 100

Step110: SAS Load the Medicaid Claims File

Step110 reads the Medicaid Claims file into a SAS data set for use in PGM400, and creates a list of Medicaid recipients who have received substance abuse and/or mental health services through Medicaid.

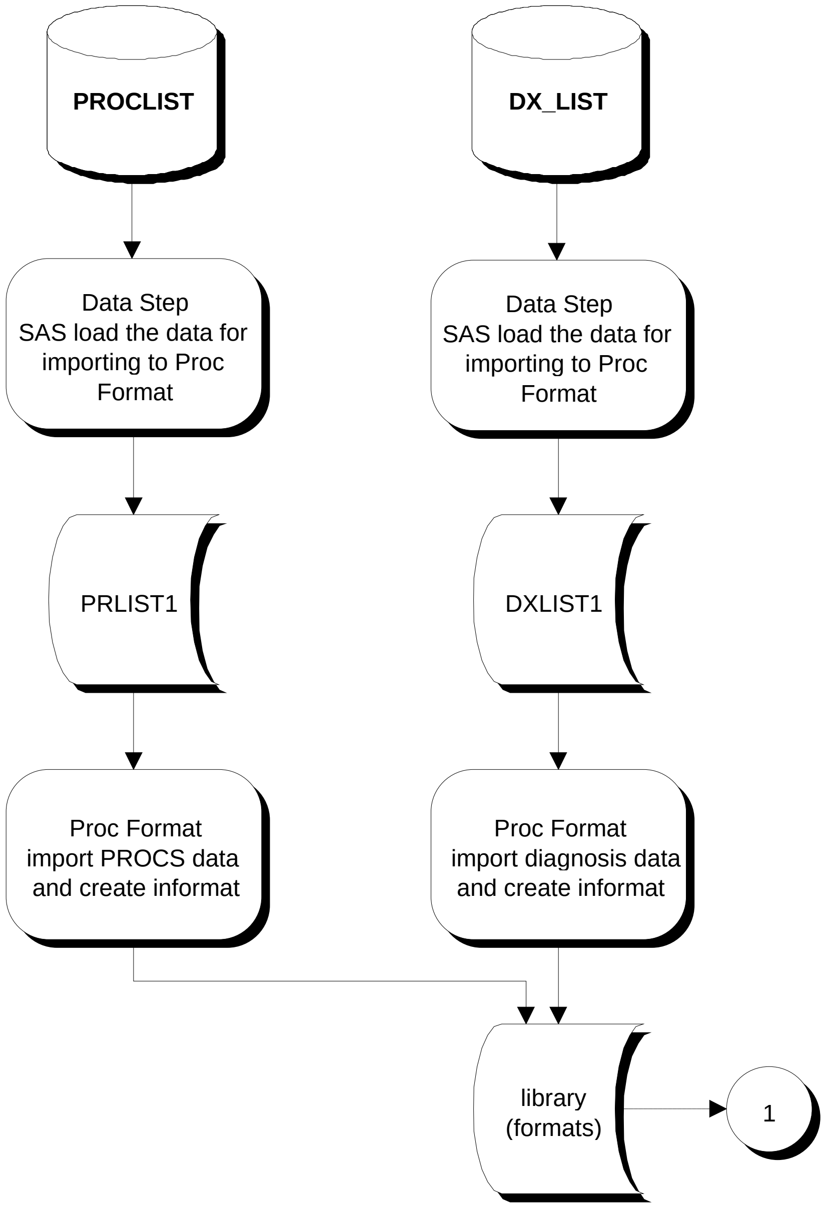

PROCLIST
DX_LIST
Data Step
SAS load the data for importing to Proc Format
Data Step
SAS load the data for importing to Proc Format
PRLIST1
DXLIST1
Proc Format
import PROCS data and create informat
Proc Format
import diagnosis data and create informat
library (formats)
1

Figure 4 - Program 100 (continued)

Step110: SAS Load the Medicaid Claims File (continued)

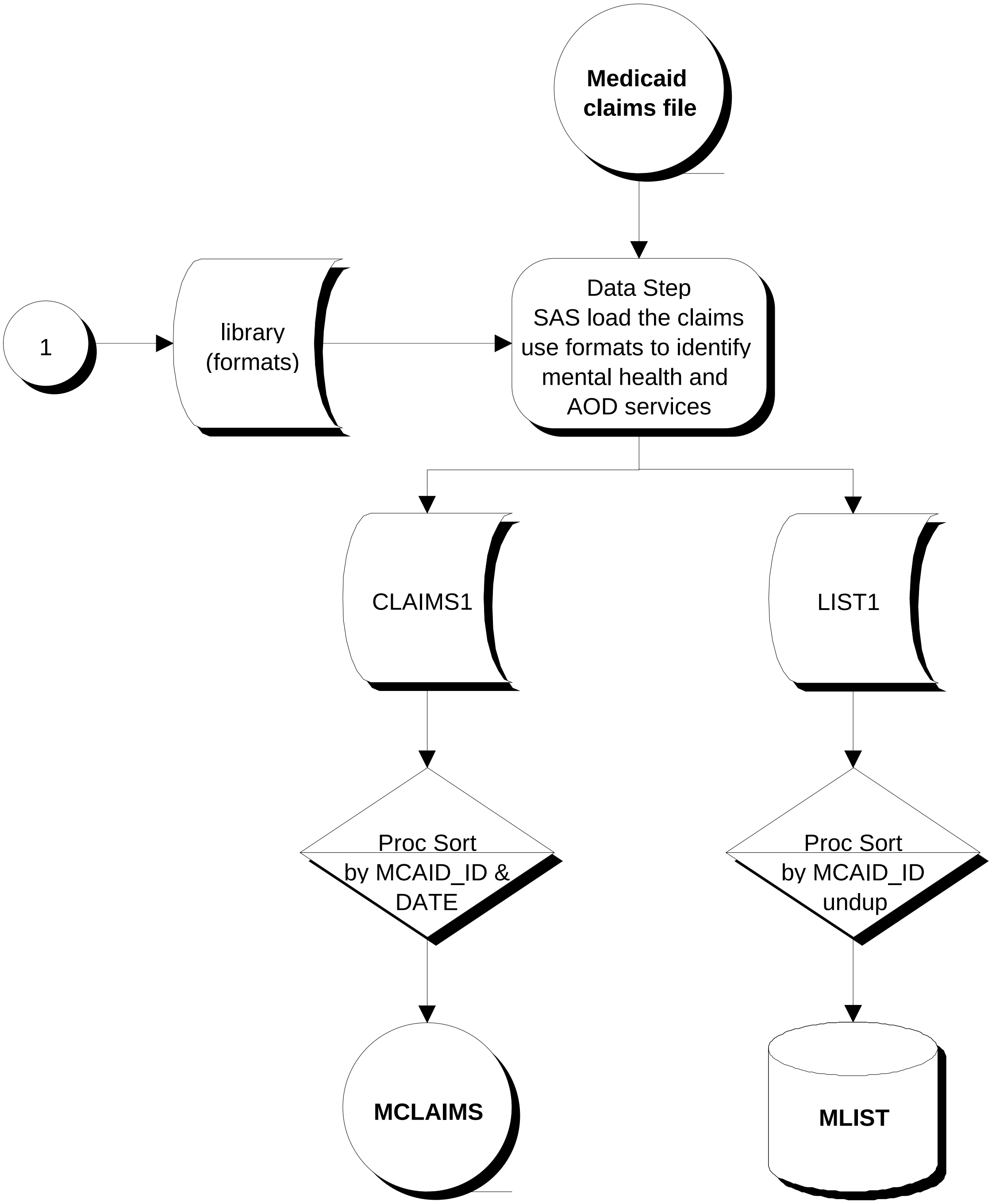


Figure 4 - Program 100 (continued)

Step120: SAS Load the Medicaid Eligibility File

Step120 reads the Medicaid Eligibility file into a SAS data set for use in PGM300. Multiple service periods (multiple records) for a recipient are loaded into date arrays and only one observation per Medicaid ID is kept in the final output data set.

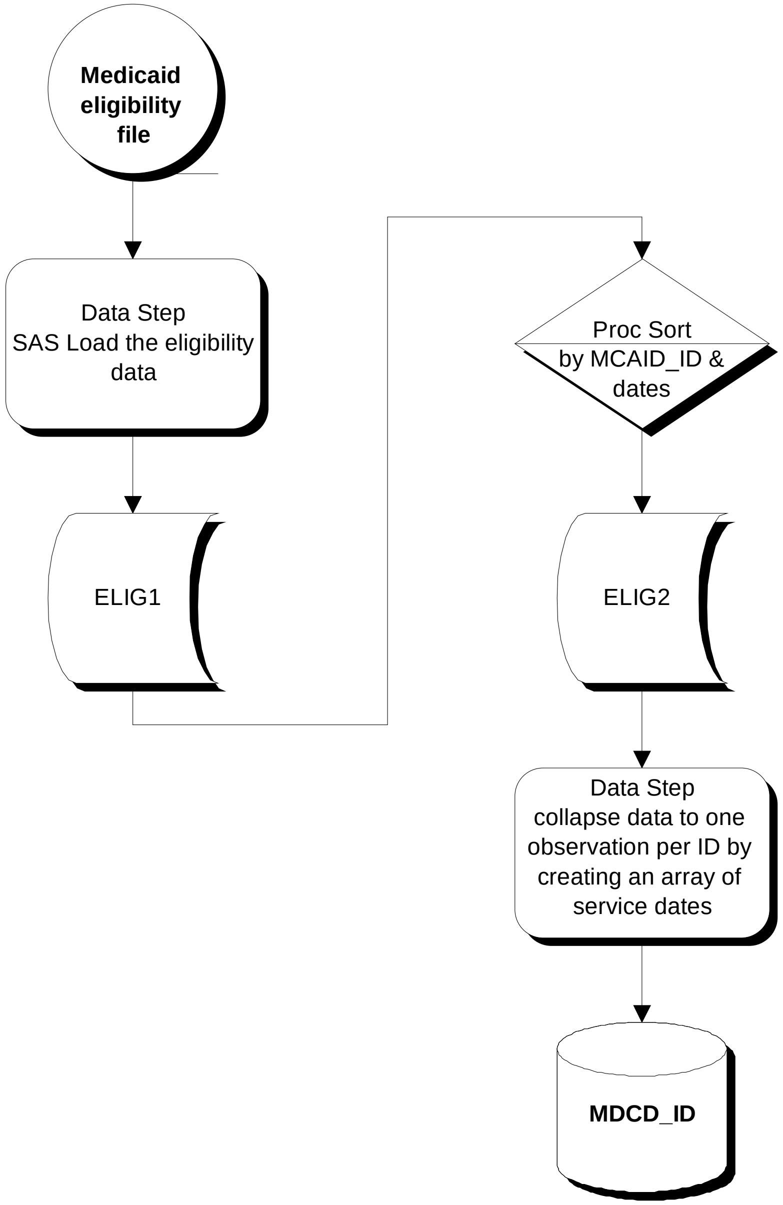
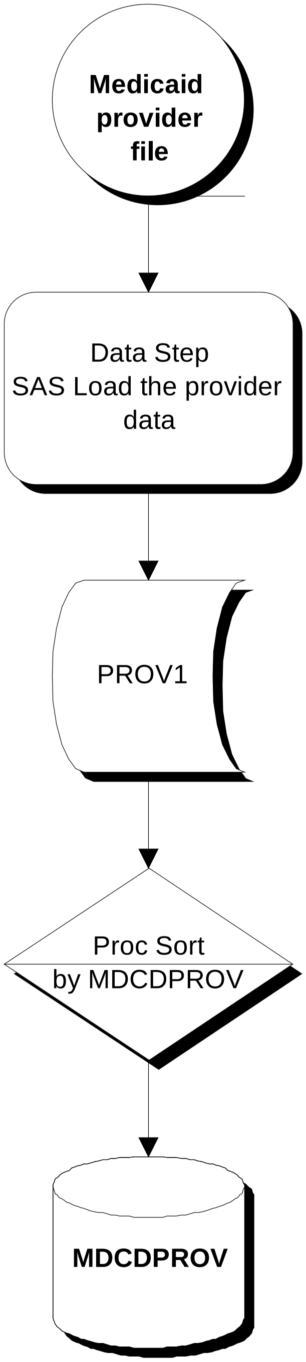

Medicaid eligibility file
Data Step
SAS Load the eligibility data
ELIG1
Proc Sort
by MCAID_ID & dates
ELIG2
Data Step
collapse data to one observation per ID by creating an array of service dates
MDCD_ID

Figure 4 - Program 100 (continued)

Step130: SAS Load the Medicaid Provider File

Step130 reads the Medicaid Provider file into a SAS data set for use in PGM500.



Program 200

Program 200 will be responsible for SAS loading the MH/AOD data from each participating state. All data files provided by the state's MH/AOD programs will be input to program 200. A separate program will be needed for each state because each state uses a unique structure for its data. Data sets created by the program will be:

- client listing - **`MHSA_ID`** - used in program 300,
- a summary of MH/AOD activity - **`MHSAACTV`** - used in program 600, and
- MH/AOD services - **`MHSASERV`** - used in program 600.

All data sets will be sorted by the identification variable assigned by the various state agencies. For this project, all such variables will be named *`STATE_ID`*.

Associating Costs with Services

We anticipate that cost information directly attributable to specific services will not be available for many MH/AOD services due to data constraints within state agencies. Cost estimation will be necessary for a number of services. This cost estimation will occur within program 200. When an actual dollar cost is included with the state's data, we will of course use that dollar amount. When cost estimation is necessary, the amount will be computed using the number of units from the data and the average cost per unit estimate. Cost per unit estimates will be derived from funding amounts in agency budgets. These will vary by state and by agency. Some budgets will allocate dollars for specific services, while other agencies may receive block funding. Every effort will be made to use budget numbers which are as detailed as possible, such as line items for a specific type of treatment rather than the agencies overall budget. Overhead costs, such as agency headquarters and community education costs, will be excluded from cost estimates, where possible.

Processing Steps

Program 200 will SAS load the MH/AOD data files. A separate, and unique program will be needed to load the data from each state in order to accommodate the different data structures and files that will be provided. Of all the programs in the processing sequence, this step will require the most customization.

A standard approach, however, will be used. This will help with programming efficiency and creating uniform data. As a general rule, data will be loaded into SAS data sets, cleaned, and sorted during the early steps of the program. Recodes, or cross-walks, to create uniform variables will take place during these steps, but no modifications to the data structure will occur. A separate step will be used for each agency that supplies data. The final step will combine all MH/AOD data and create three SAS data sets of

- service level data, where available, providing detailed information on each MH/AOD service,
- summarized activity, and
- ID/demographics data with one observation for each state ID.

Any and all service level data provided by the states will be retained. Unfortunately, due to data limitations of some participating states agencies, service level data is not always available.

Figures 5, 6, and 7, beginning on pages 22, 26, and 29 show the processing stream for Delaware, Oklahoma, and Washington data, respectively.

Figure 5 - Program 200, Delaware

Step210: SAS Load the Adult MHSA Files: Clients, Inpatient MH, Outpatient MH, and AOD

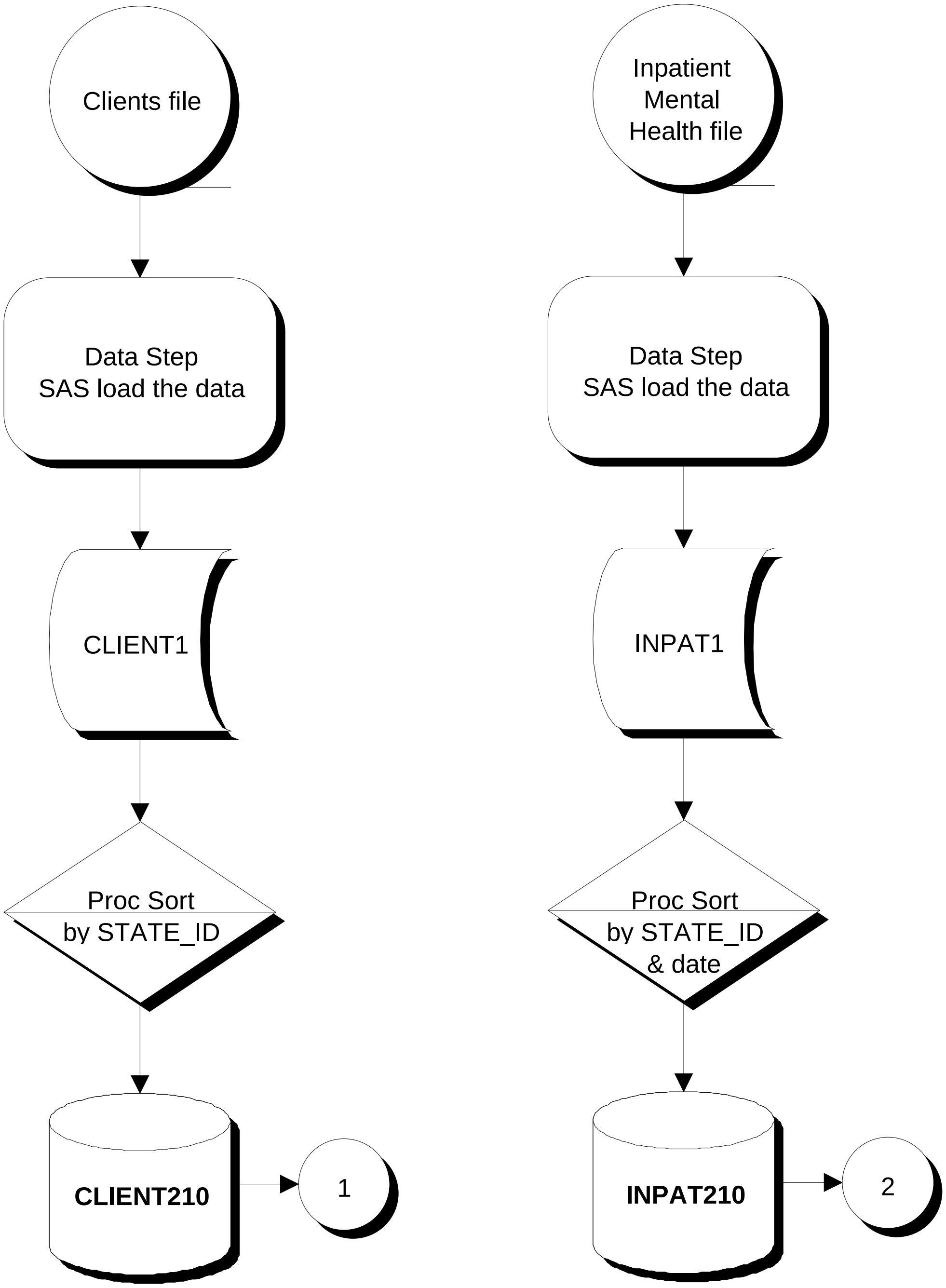

Clients file
Inpatient Mental Health file
Data Step SAS load the data
Data Step SAS load the data
CLIENT1
INPAT1
Proc Sort by STATE_ID
Proc Sort by STATE_ID & date
CLIENT210
1
INPAT210
2

Figure 5 - Program 200, Delaware (continued)

Step210: SAS Load the Adult MHSA Files: Clients, Inpatient MH, Outpatient MH, and AOD

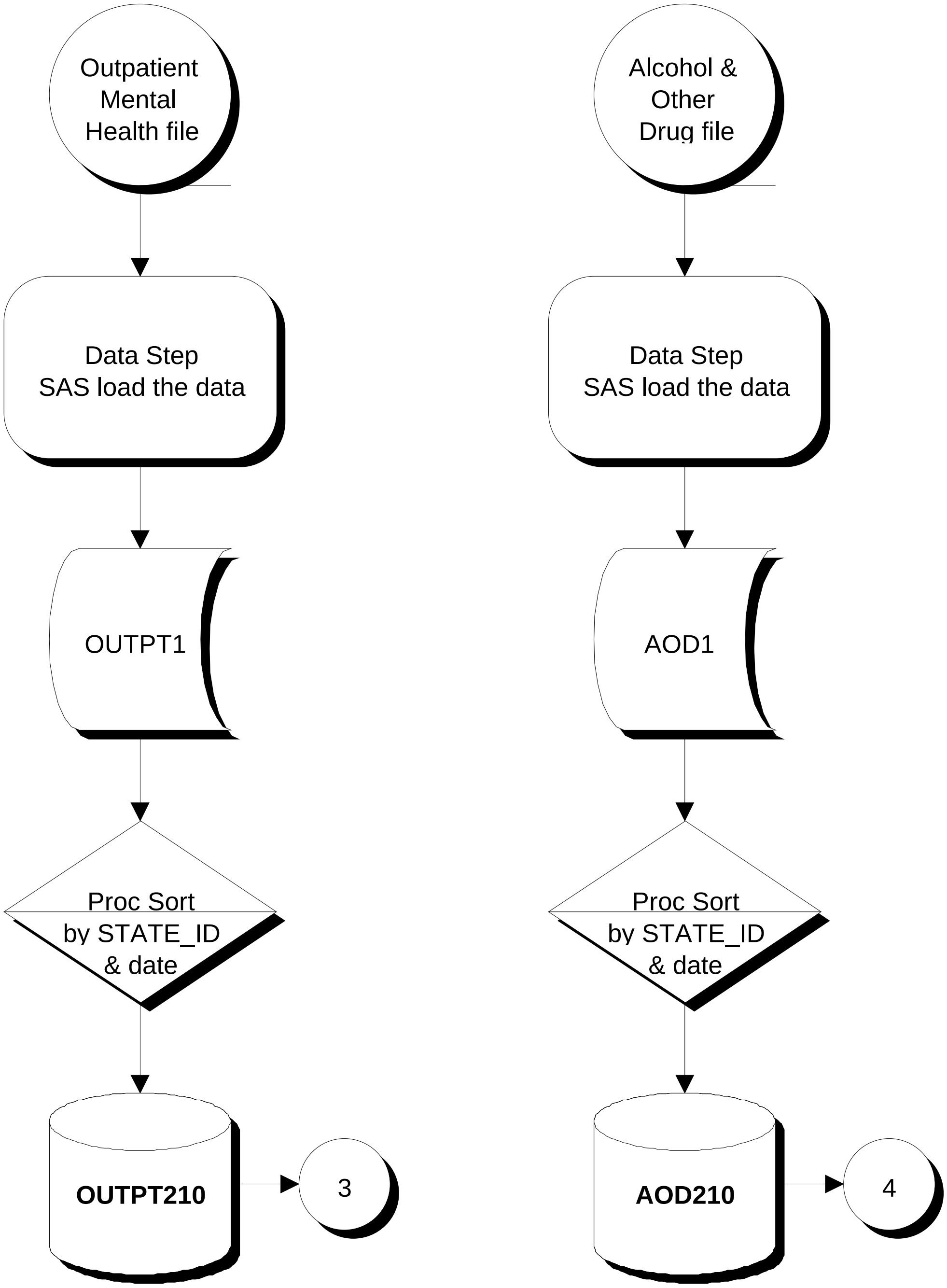

Outpatient Mental Health file
Data Step SAS load the data
OUTPT1
Proc Sort by STATE_ID & date
OUTPT210
3
Alcohol & Other Drug file
Data Step SAS load the data
AOD1
Proc Sort by STATE_ID & date
AOD210
4

Figure 5 - Program 200, Delaware (continued)

Step220: SAS Load the Division of Childeren and Youth Files: Clients and Services

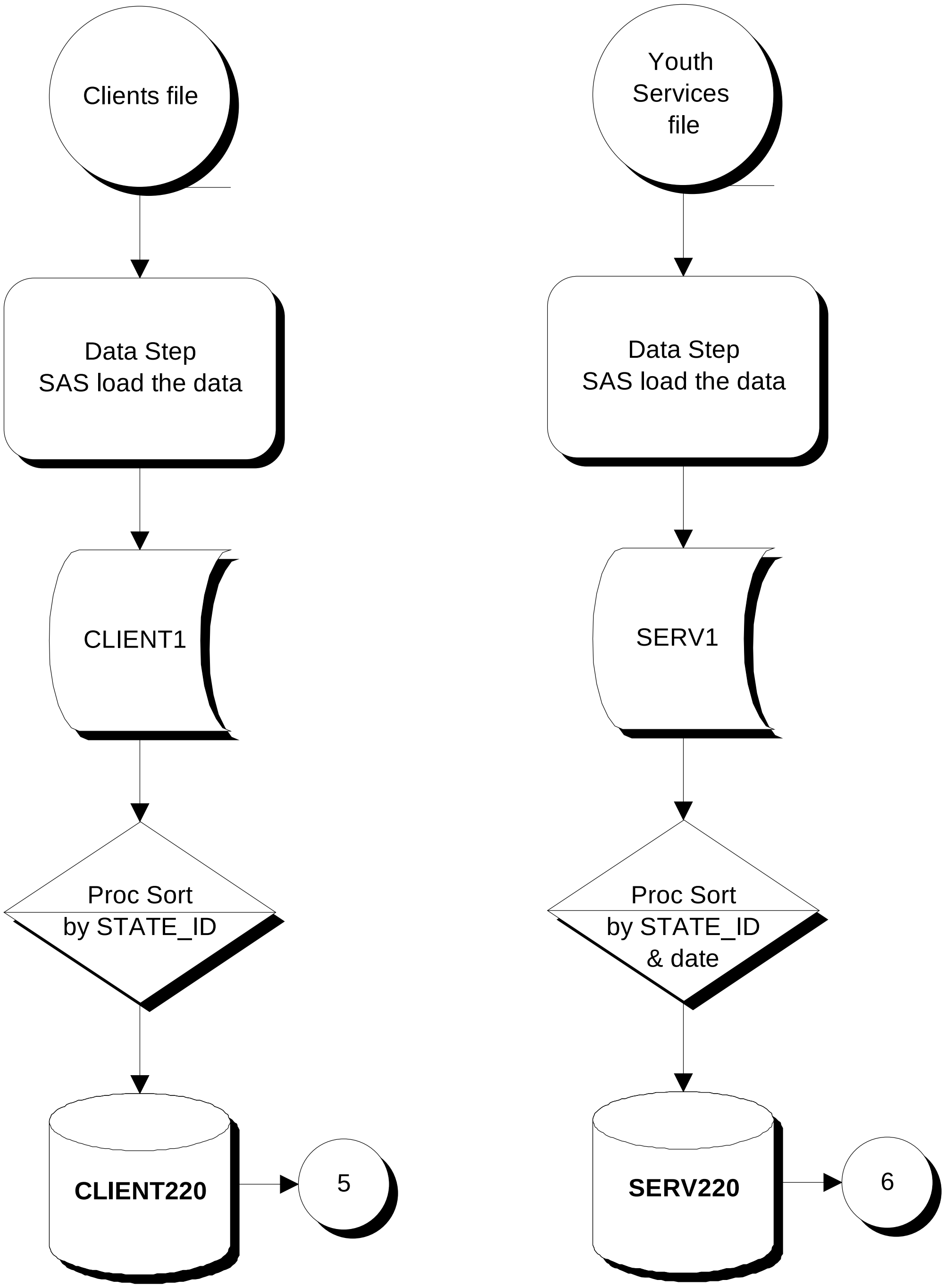

Clients file
Youth Services file
Data Step SAS load the data
Data Step SAS load the data
CLIENT1
SERV1
Proc Sort by STATE_ID
Proc Sort by STATE_ID & date
CLIENT220
5
SERV220
6

Figure 5 - Program 200, Delaware (continued)

Step230: Combine the MH/AOD Source Data and Create Demographics and Services Data Sets

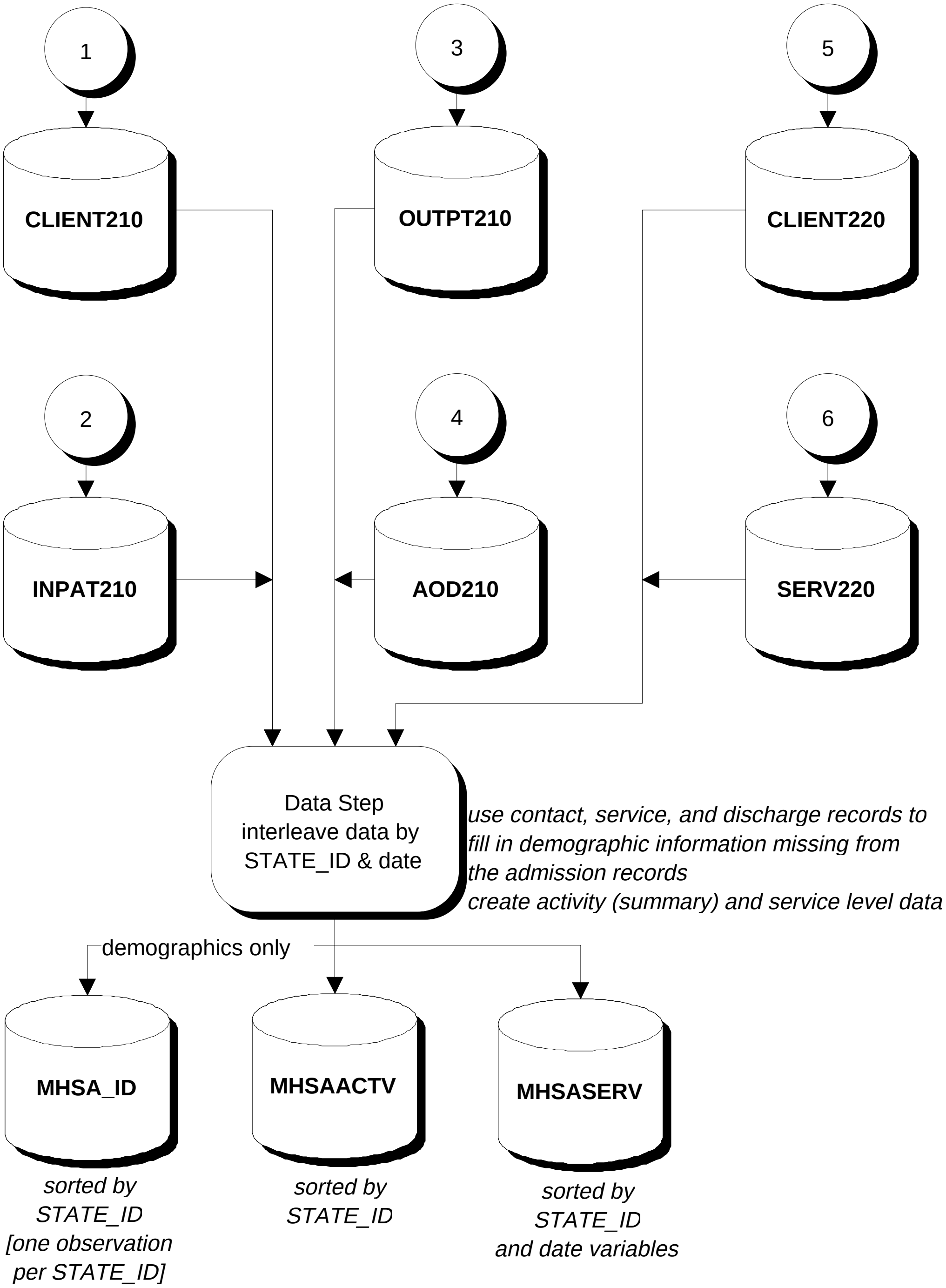


Figure 6 - Program 200, Oklahoma

Step210: SAS Load the MH/AOD Files: Admissions, Services, Discharges, and Contacts

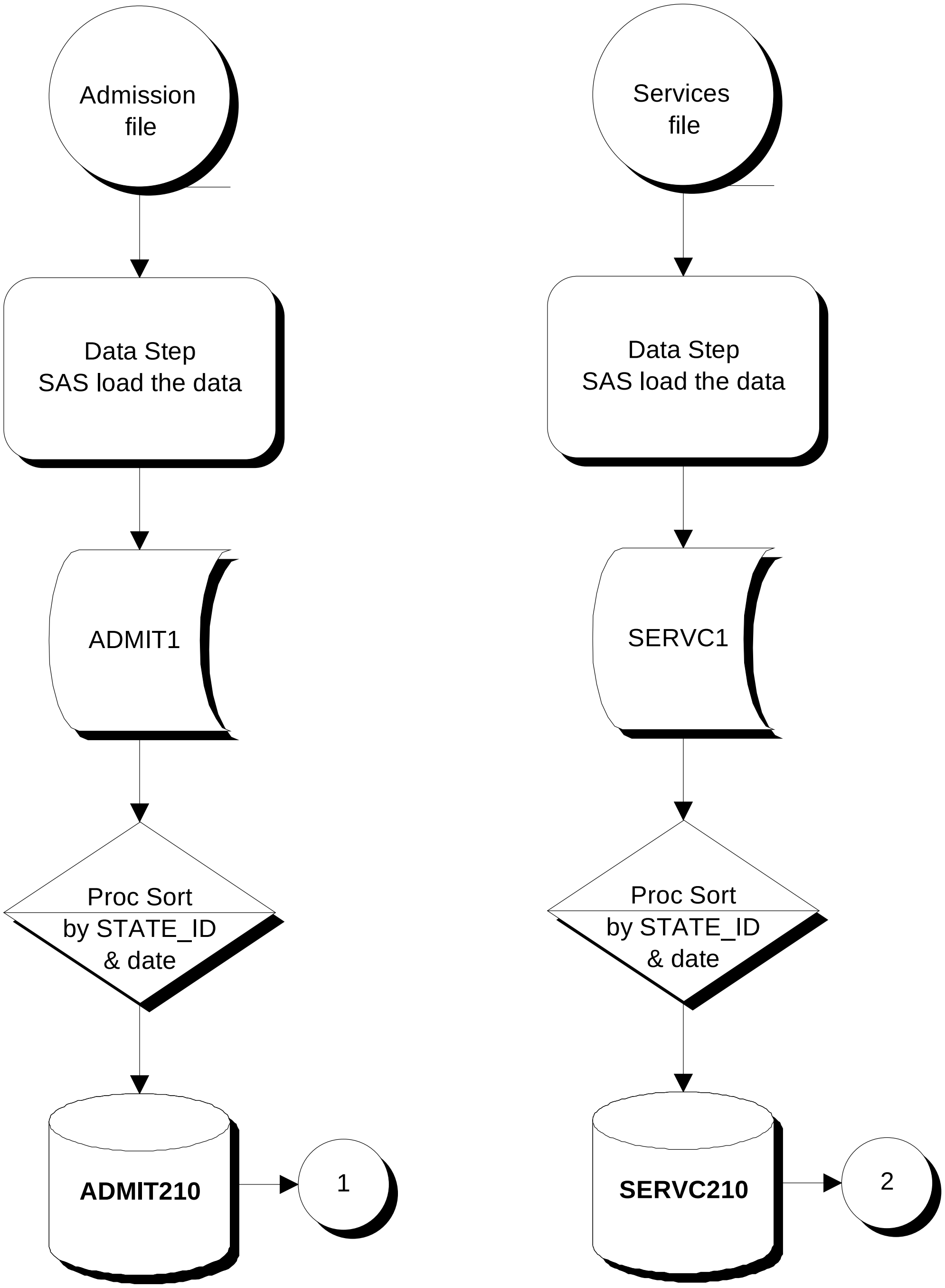

Admission file
Services file
Data Step SAS load the data
Data Step SAS load the data
ADMIT1
SERVC1
Proc Sort by STATE_ID & date
Proc Sort by STATE_ID & date
ADMIT210
1
SERVC210
2

Figure 6 - Program 200, Oklahoma (continued)

Step210: SAS Load the MH/AOD Files: Admissions, Services, Discharges, and Contacts (continued)

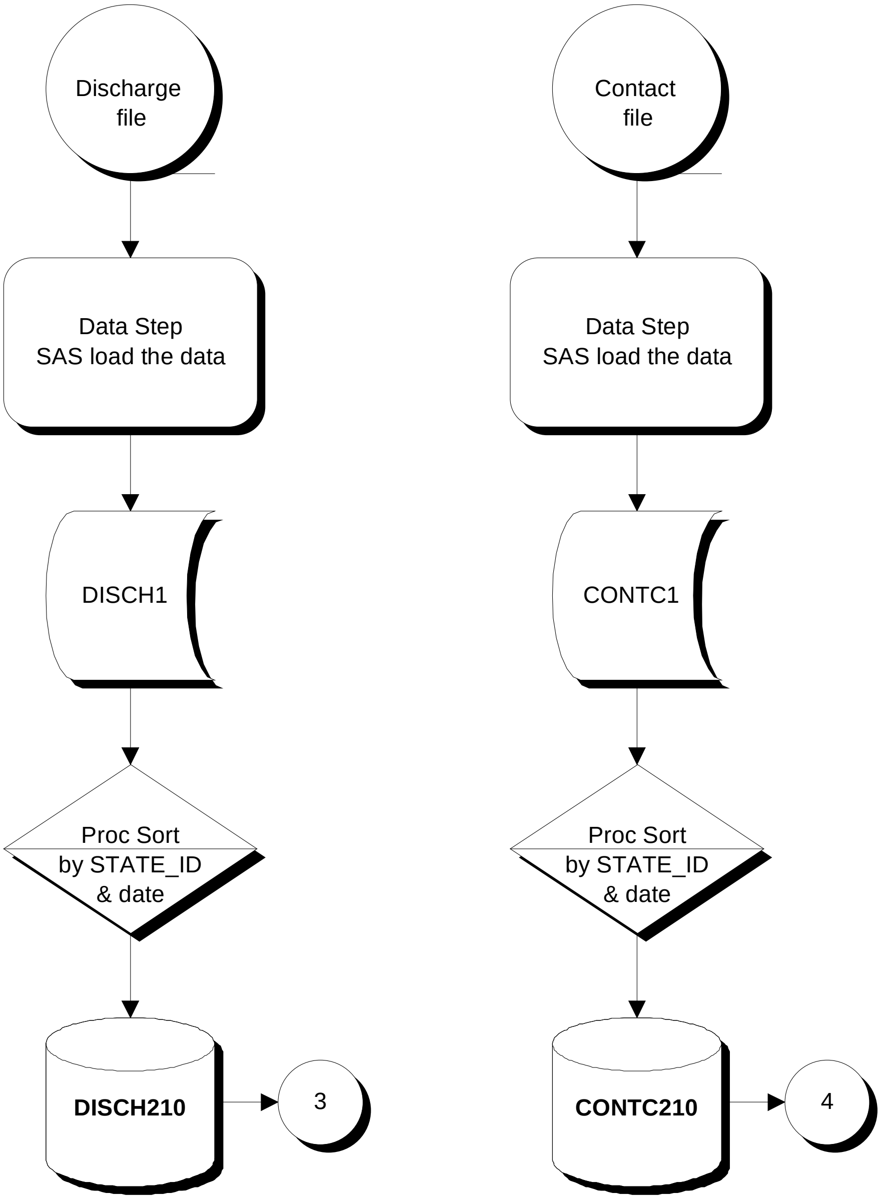

Discharge file
Contact file
Data Step SAS load the data
Data Step SAS load the data
DISCH1
CONTC1
Proc Sort by STATE_ID & date
Proc Sort by STATE_ID & date
DISCH210
3
CONTC210
4

Figure 6 - Program 200, Oklahoma (continued)

Step220: Combine the MH/AOD Source Data and Create Demographics and Services Data Sets

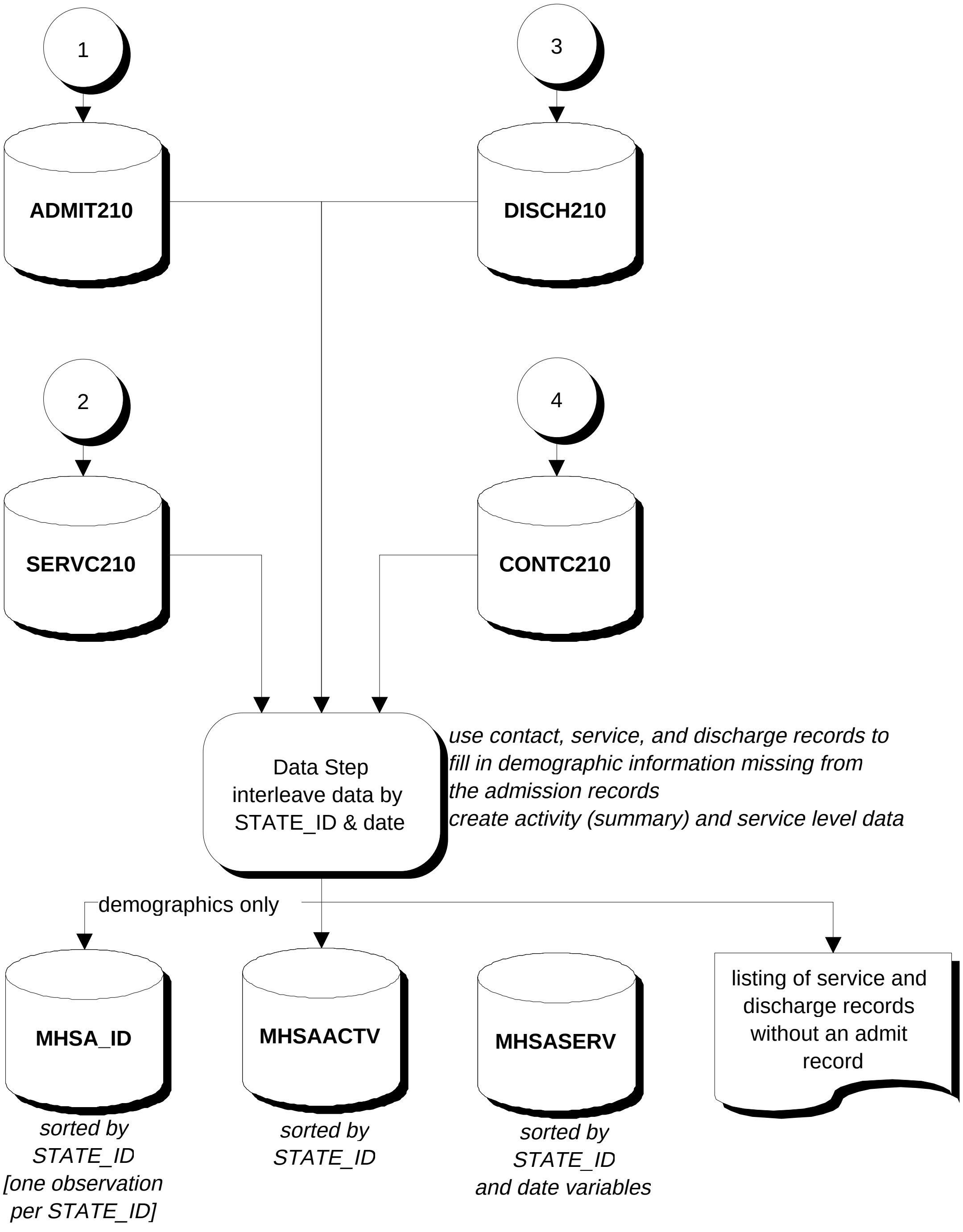


Figure 7 - Program 200, Washington

Step 210: SAS Load the DASA Data

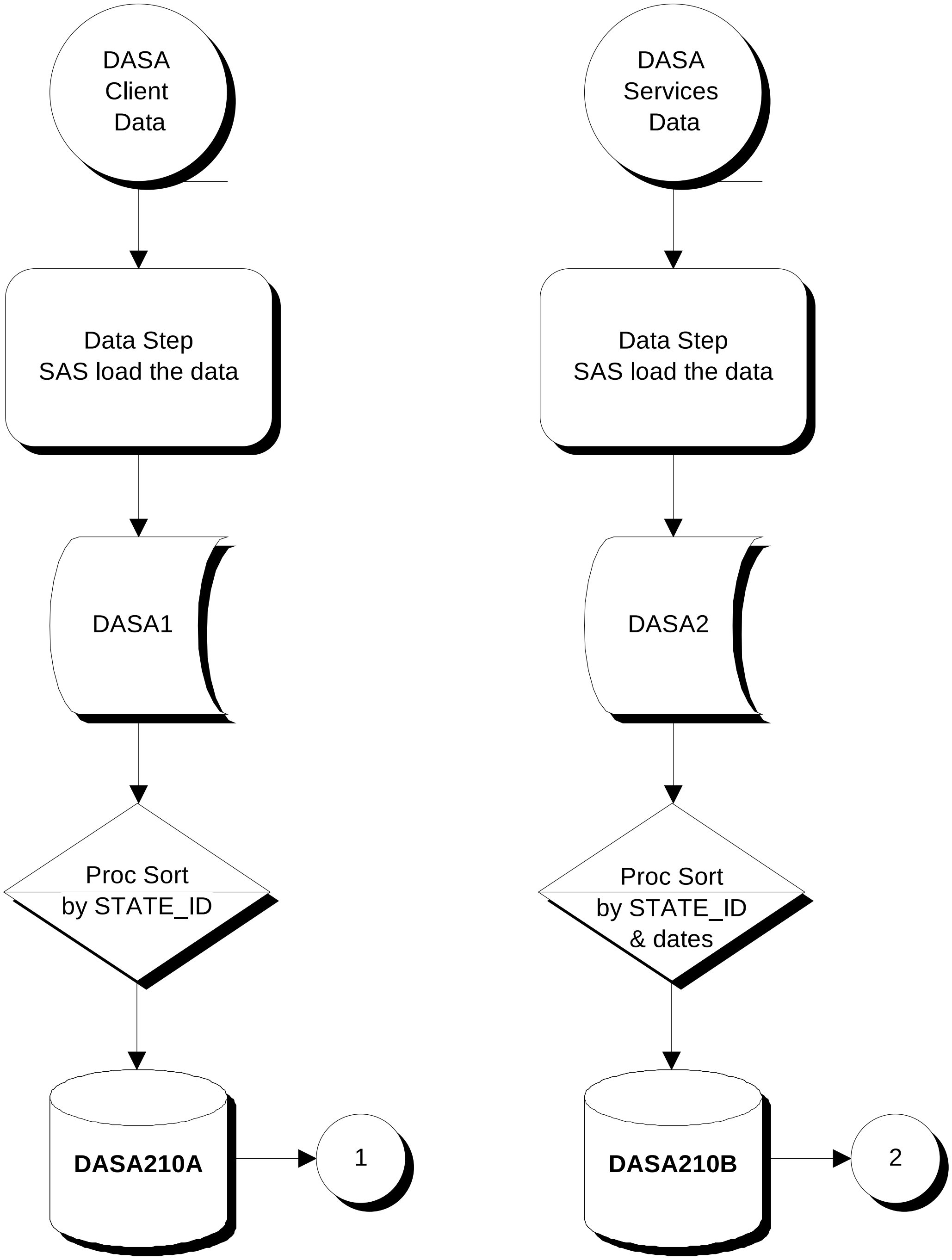


Figure 7 - Program 200, Washington (continued)

Step 220: SAS Load the Mental Health Data

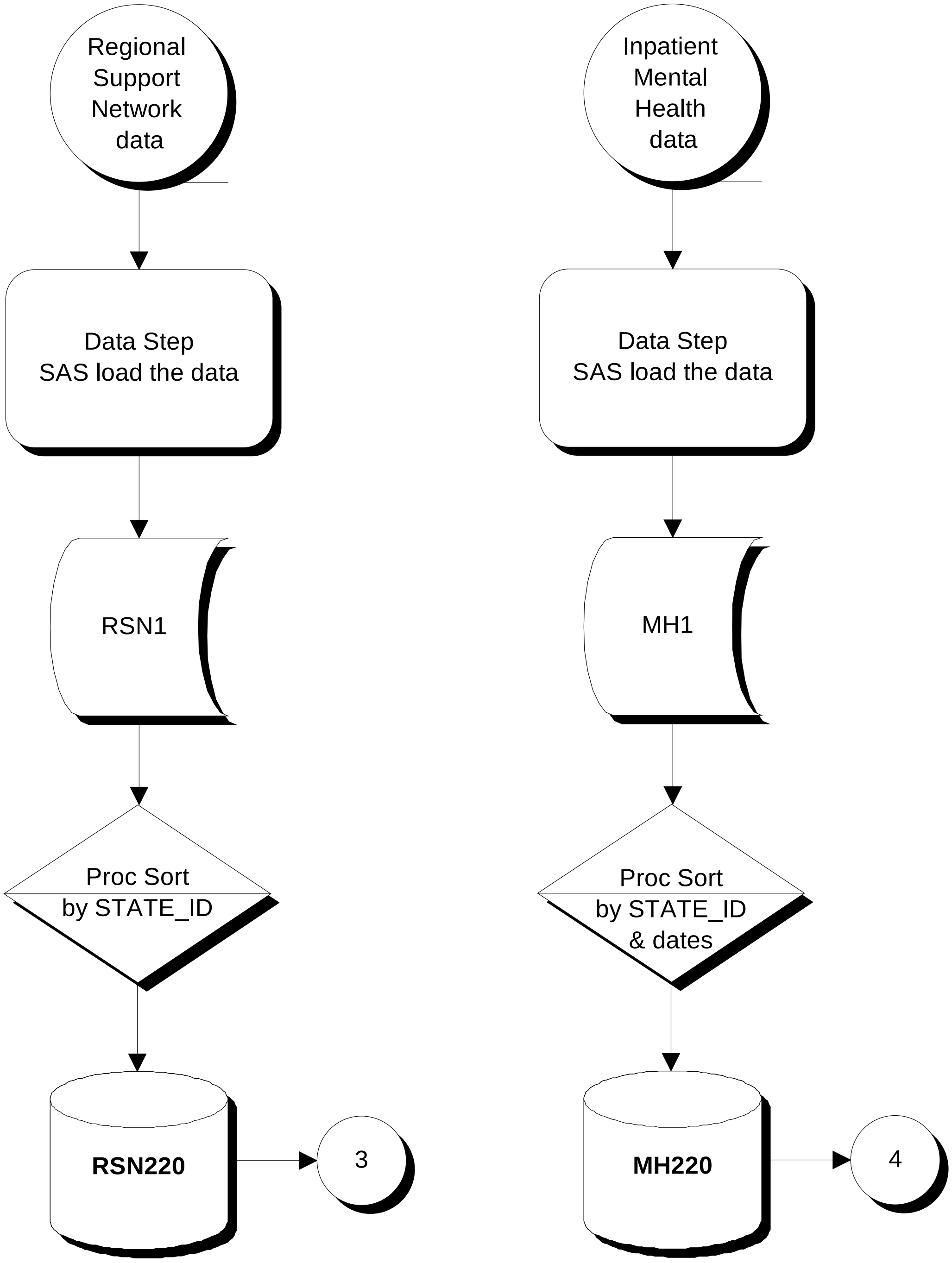
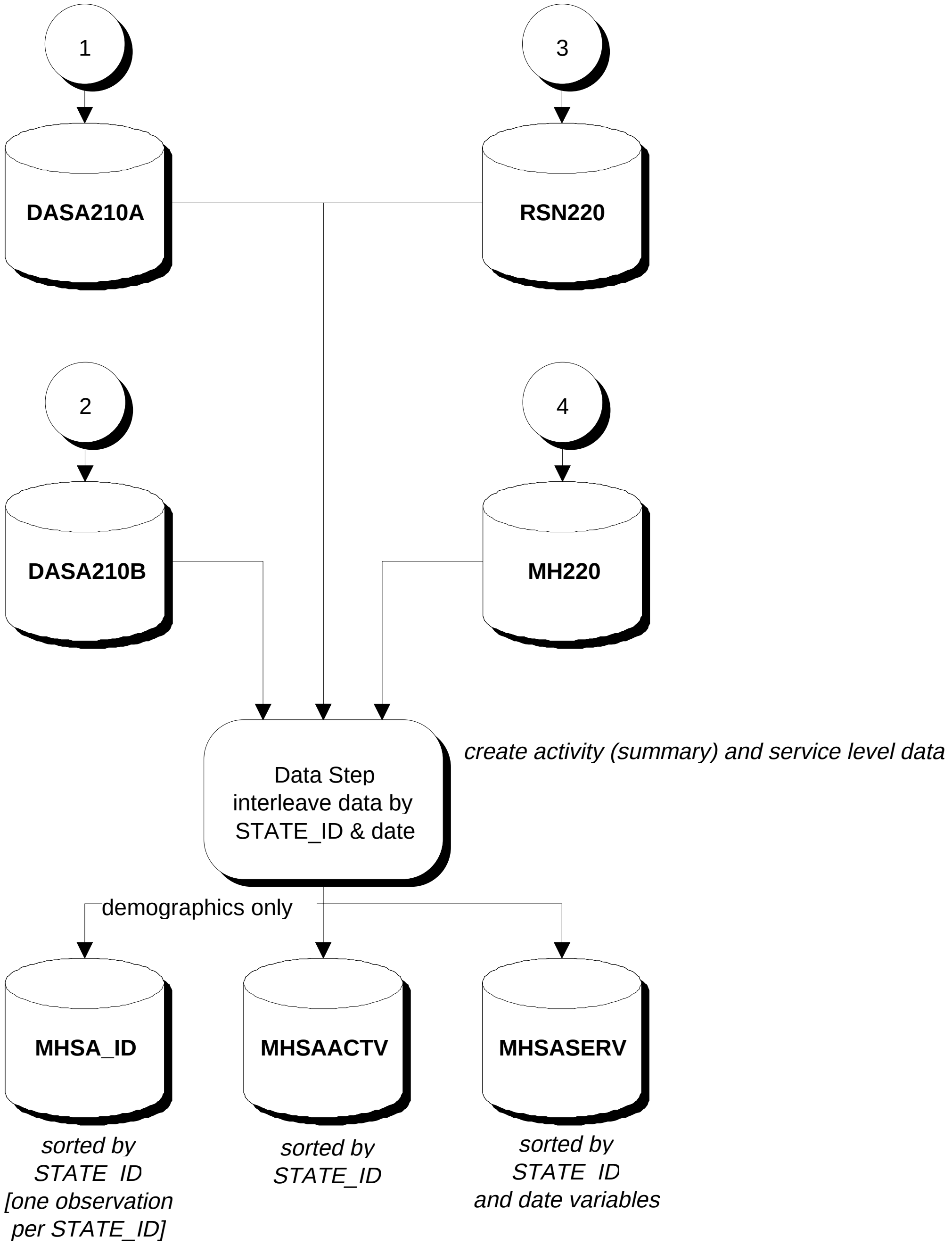

Regional Support Network data
Data Step SAS load the data
RSN1
Proc Sort by STATE_ID
RSN220
3
Inpatient Mental Health data
Data Step SAS load the data
MH1
Proc Sort by STATE_ID & dates
MH220
4

Figure 7 - Program 200, Washington (continued)

Step 230: Combine the DASA and Mental Health Data and Create Demographics and Services



Program 300

The third program in the sequence, program 300, will process identification and demographic data, and create a “master” ID data set: **IDMASTER**. **IDMASTER** will map Medicaid and non-Medicaid IDs (*MCAID_ID* and *STATE_ID*) to a common, project assigned ID: *CLIENT_ID*. The process of assigning *CLIENT_ID*s will be described later in the “UNDUP Include” section. ID mapping will be an iterative process: mappings determined in prior processing runs will serve as the starting point for mappings in new processing runs. Using an iterative process will allow us to repeat the process with additional data and still maintain consistency in mapping the ID links. This will be necessary if a complete year of data is not initially available from any of the states, or if additional years of data are added to the project. The **IDMASTER** data set created during the previous execution of program 300 will be an input to the new program 300. During the first iteration of program 300, no **IDMASTER** data set will be available, so a special data set containing *zero* observations will be created as a place holder. **IDMASTER** will therefore be both an input to and an output from the program. The master ID data set, **IDMASTER**, will contain all mappings from *CLIENT_ID* to both *MCAID_ID* and *STATE_ID*. **IDMASTER** will be sorted by *CLIENT_ID* with indexes created on the variables *MCAID_ID* and *STATE_ID*.

The Medicaid ID data set, **MDCD_ID**, from program 100 and the MH/AOD ID data set created in program 200, **MHSA_ID**, will be the primary inputs to program 300. An additional input will be a small data set containing the number of the last *CLIENT_ID* number assigned. This will be needed to track *CLIENT_ID* and ensure that each is unique.

Linking and Unduplicating Clients

Linking Issues

In order to build the integrated database, clients from two different sources, MH/AOD and Medicaid, must be linked. There are a number of issues involved with linking client records that complicate the linking process. If a universal identifier were available, linking clients would be a simple matter. Unfortunately, no universal identifier is available so client linking must be performed using other identifying information, where available, such as:

- Medicaid ID,
- Social Security Number,
- Name (first and last),

- Socio-demographic information (i.e. gender, race, and ethnicity),
- Client's date of birth, and
- Client's home ZIP code.

We will use all available identifying information when linking client records because, for numerous reasons, identifying information is often unavailable or not completely reliable. In some cases identifying information is not recorded in an effort to protect the client's privacy. Many patients, fearing that their privacy will be compromised, are unwilling to provide accurate information when receiving MH/AOD services. For some, the worry is social stigmatization; others fear increased scrutiny from law enforcement or the loss of their children to Social Services. Identifying data, needed for linking these client's records, is often missing as a result of these fears. Social Security Numbers, for example, are frequently blank on AOD data even though clinics which receive Federal funds are allowed to collect Social Security Numbers.

Even when identifying information is available, it is frequently subject to errors and not completely reliable. This may be due to recording errors or because clients are unable to supply accurate information as a result of their condition. Clients entering a detox program are often incapable of providing their name, or they furnish a pseudonym. Problems and errors can arise even when clients are willing and able to supply information. For example, digits may be transposed in ID numbers and birth dates. Names also present linking problems for a variety of reasons, such as:

- spelling (e.g. Smith, Smyth, and Smythe),
- nicknames (e.g. Elizabeth, Beth, Betsy, and Liz), and
- transposing First and last names (especially common with Asian names).

Because of these issues, the linking of client records will be rather complex. Particular care and effort will be paid to cleaning identifying data, particularly names which will be standardized as much as possible. Data will be combined using several different criteria. We will look for matches on all identifying variables, and will look beyond exact matches for a number of those variables.

There are two types of errors that can result from the linking process, which we will call *False Negative* errors and *False Positive* errors. *False Negative* errors occur when client observations which should be linked remain separated. This is likely to occur when identifying data is bad or missing. The other type of error, *False Positive* errors, occur when observations for two different clients are mistakenly linked. Obviously, we would like to minimize both types of errors. However, *False Negative* and *False Positive* errors are related in that efforts to minimize one increases the chances of the other. For example, *False Positive* errors are less likely with very stringent the linking criteria. But the stringent criteria will increase the number of *False Negative* errors. We have designed the linking process to allow for flexibility in choosing the balance between these types of errors.

The Linking and Unduplication Process

Linking and Unduplicating is actually a multi-step process involving:

1. unduplicating clients from the base data (Medicaid),
2. linking clients from the two sources (Medicaid and MH/AOD),
3. unduplicating the clients linked above, and
4. unduplicating the remaining clients from the second source (MH/AOD) - these are the client IDs which did not link to the base data in item 2.

Unduplicating clients from a single data set source (numbers 1 and 4), and linking clients from two different data sets (number 2) are conceptually similar. On a general level, both match each observation with all other observations, using a SQL join, and creating the Cartesian[5] product of the data. Unduplicating clients matches observation within a single data set to find duplicates (a reflexive join). Linking clients from two data sets matches each observation from a unduplicated data set with every observation from the second data set (an inner join). Unduplicating one of the data sets before linking is a necessary step before linking. In either case, each matched pair of observations is scored to determine the likelihood that the two observations identify the same client. The higher the score, the greater the prospect that the two observations indeed represent one client. The next section, “Scoring Links” explains the scoring criteria.

[5] The Cartesian product contains *every* combination of rows from the joined tables, or data sets. As an example, the Cartesian product from the join of a data set of 1000 observations and another data set of 2000 observations, is a new data set with 2 million (1000 X 2000) observations.

Scoring Links

Scoring the links serves two purposes. It enables us to discard matched observations that clearly do not represent one client by setting a threshold, or minimum score required for the match. A low threshold will help reduce *False Positive* errors, since match scores must exceed the threshold to be considered. More importantly, scoring identifies the best link for each client. Each observation will initially be matched with many other observations. Even after eliminating matches where the score is below the threshold, it is likely that a client observation will be matched with several other observations. The best match will be identified by the scores: the highest score for each client will be retained.

Figure 8 shows our initial criteria for scoring matches. Point values are for illustrative purposes to serve as an example of how scores will be awarded. In practice, a considerable amount of testing will be necessary to determine the optimal scoring methods. Scoring will be performed on all identifying variables to reduce the number of *False Negative* errors. Points will be added for variable matches, either complete or partial, and removed for mismatches. Partial matches receive points to accommodate coding errors. If the variable is not available on at least one of the observations, no points will be added or subtracted.

Figure 8 - Scoring of Matches/Joins

	complete	partial	no-match	no-match
Variable	match	match*	(missing)	(unequal)
First ID (PIC or SSN)	25	10 to 20	0	-10
Second ID (SSN or PIC)	25	10 to 20	0	-10
Date of Birth	20	5 to 10	0	-3
Gender	10	0	0	-1
Name	35	5 to 25	0	-5
Race	5	0	0	-1
Ethnicity	5	0	0	-1
ZIP Code	5	3	0	-1

* Partial matches:

IDs, match on all characters except one, 20 points
match on all but two characters, 18 points
match on all but three characters, 14 points
match on all but four characters, 10 points
mis-match on more than four characters, -10 points

Figure 8 - Scoring of Matches/Joins (continued)

* Partial matches (continued):

DOB: off by one day, 10 points
off by one year, 6 points
same year/month, one date is first of month, 8 points
same year/month, one date is NOT first of month, 5 points
same year, one date is January 1st, 7 points

Name [Last-name match / First-name match]:
first occurrences / any order match, 25 points
first occurrences / match of initials, 20 points
first occurrences / no match, 15 points
any order match / first occurrences, 25 points
any order match / any order match, 22 points
any order match / match of initials, 17 points
any order match / no match, 12 points
match of initials / first occurrences, 12 points
match of initials / any order match, 9 points
match of initials / match of initials, 7 points
match of initials / no match, 5 points

[First- to Last-name match / First- to Last-name match]
first occurrences / first occurrences, 20 points
first occurrences / any order match, 15 points
any order match / first occurrences, 15 points
any order match / any order match, 10 points

ZIP: match for postal center (first three digits), 3 points

Optimizing Links

Creating the Cartesian product of the data can produce massively large data sets: billions of observations are possible. In practice it will not be necessary to produce the complete Cartesian product, because we will want a match on at least two of four main identification variables:

1. Medicaid ID (PIC code),
2. Social Security Number
3. Date of Birth and Gender, or
4. First and Last Name.

Linking will be optimized with separate joins on the above criteria. Because we will want matches on at least two of these variables, we only need to create joins on three of the criteria. Using a SQL clause requiring the specified variables from each joined observation to be equal, the data sets will be much smaller than the complete Cartesian product (although still quite large). Joins that use an equality clause such as this are referred to as an equi-joins. Besides creating smaller data sets, equi-joins generally require less processing time.

Processing Steps

The purpose of program 300 is to link and unduplicate clients, and assign client IDs, as described in the previous sections. Figure 10 beginning on page 40 shows the flowchart for program 300. The linking and unduplicating process is complicated, and the numerous files and variables make the descriptions quite involved. In order to facilitate an understanding of the proposed steps, we provide definitions in Figure 9 (on page 38) for the files and variables referenced in the following discussion.

The first step (310) will unduplicate the Medicaid ID data set, **`MDCD_ID`**, using the UNDUP include. In a later section we will discuss the details of UNDUP, which is a file of generalized unduplicating code that is *included* within other programs - namely program 300. UNDUP is the section of code where *`CLIENT_ID`* will be assigned. For the linking process, the Medicaid data, **`MDCD_ID`**, will serve as the base data. It is our belief that the Medicaid data will contain more accurate and a more complete set of identifying variables than the MH/AOD data, and will therefore produce better linking results. The first linking task will unduplicate the base, or Medicaid, data using UNDUP . Before running the data set **`MDCD_ID`** through UNDUP, the Medicaid data will first be merged with the ID mapping data set created during the previous execution of program

300: **IDMASTER**. (**IDMASTER** will be an empty data set for the first iteration of the process). Merging with the "old" **IDMASTER** will add any previously found socio-demographic information not contained on the new data sets, and assure consistent ID mappings over time.

Figure 9 - ID Variables and Files Used in Linking and Unduplicating Clients

ID Variables:	
CLIENT_ID	unique ID assigned in program 300 to identify linked and unduplicated client records
MCAID_ID	Medicaid ID
STATE_ID	ID from the state MH/AOD agencies (*may originate from more than one agency*)
Files:	
IDMASTER	data set containing all *CLIENT_ID*s and mappings to *STATE_ID*s and *MCAID_ID*s with demographic information. Includes the ID variables *CLIENT_ID*, *MCAID_ID*, and *STATE_ID*.
MDCD_ID	Medicaid ID/demographics data set from program 100. Includes the ID variable *MCAID_ID*.
MHSA_ID	State MH/AOD data set with ID and demographics information from program 200. Includes the ID variable *STATE_ID*.

The second step (320) will combine the Medicaid and MH/AOD ID data, link, and unduplicate clients. MH/AOD ID data from program 200 will first be merged with the "old" **IDMASTER** data set to add any socio-demographic missing from the new MH/AOD data. The Medicaid and MH/AOD data will then be combined, using SQL equi-joins using the three criteria described above in the section "The Linking and Unduplication Process". Those three criteria are:

- Medicaid IDs are equal,
- Social Security Numbers are equal, and
- both gender and date of birth are equal.

After each join, the matched records will be scored, as described in the prior section "Scoring Links", and sorted by *STATE_ID*. Data from the three equi-joins will then be combined. For each group of *STATE_ID*s, the match with the highest score will be used as the *STATE_ID* to *CLIENT_ID* mapping.

Step 320, by linking MH/AOD IDs to Medicaid IDs, will assign a *CLIENT_ID* mapping to *STATE_ID*s. But these mappings will only occur for MH/AOD clients in the Medicaid program. Additional mappings will be needed for MH/AOD clients that are not in the Medicaid program. These mappings will be made in step 330 with the UNDUP include.

The final step (340) combines data created in the three other steps. The unduplicated Medicaid data from step 310 and the linked data from step 320 will be merged by Medicaid ID to update demographic variables and add *STATE_ID* mappings. This data set will then be combined with the unduplicated data from step 330 and sorted by *CLIENT_ID*. Indexes will be created on *MCAID_ID* and *STATE_ID*, creating the new master ID data set: **IDMASTER**.

Figure 10 - Program 300

Step310: De-duplicate the IDs from the Medicaid Enrollment Data

Step 310 de-duplicates the Medicaid clients. ID links from the previous processing sequence are added t(the Medicaid ID data and updates to demographic information is made. The UNDUP include file is called, setting FileID to "MEDCD_ID" and MAP_ID to "CLIENT_ID".

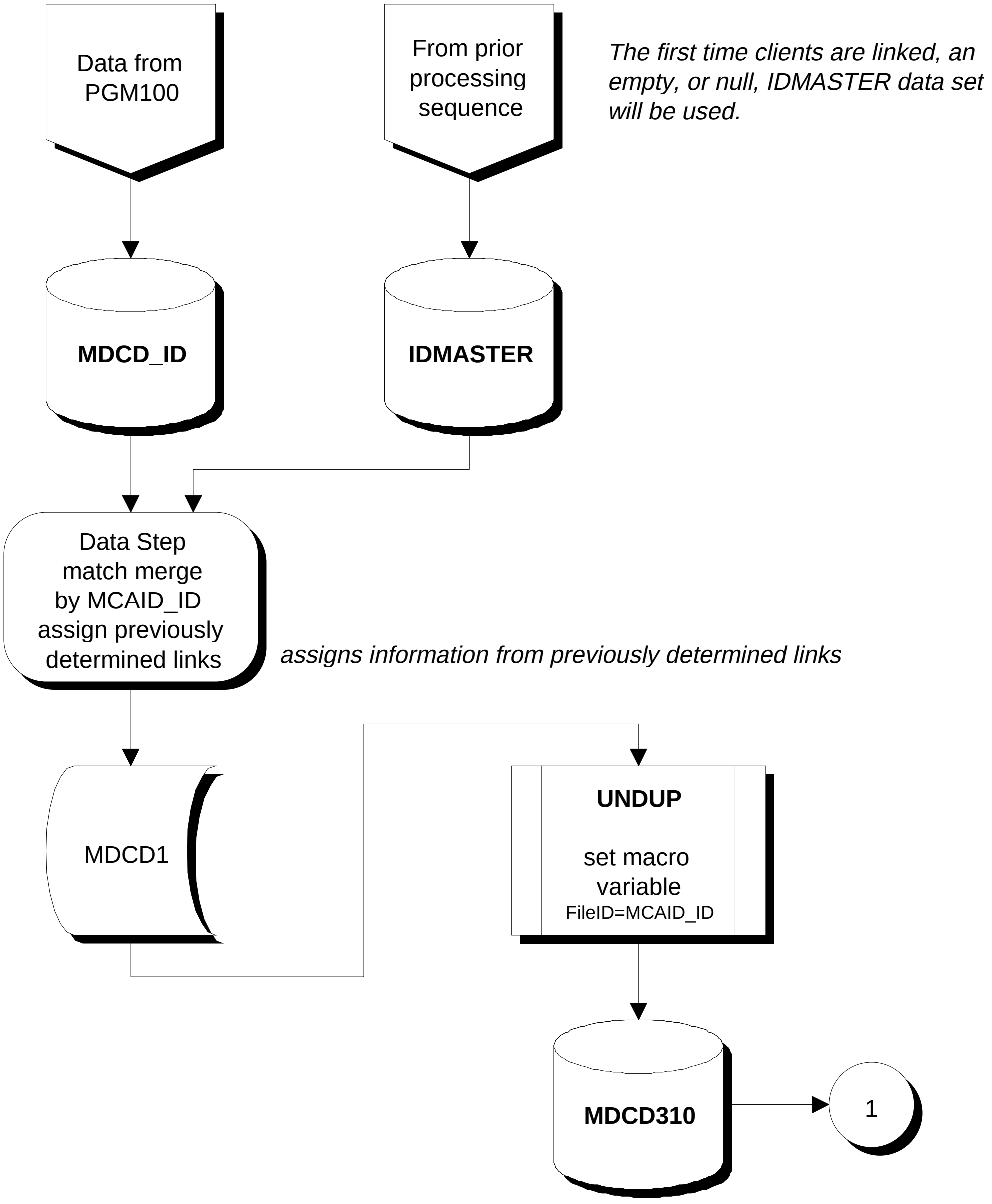

Data from PGM100
From prior processing sequence
The first time clients are linked, an empty, or null, IDMASTER data set will be used.
MDCD_ID
IDMASTER
Data Step match merge by MCAID_ID assign previously determined links
assigns information from previously determined links
MDCD1
UNDUP
set macro variable
FileID=MCAID_ID
MDCD310
1

Figure 10 - Program 300 (continued)

Step320: Matches Medicaid and MH/AOD Data

Step320 combines the Medicaid ID data with the Substance Abuse/Mental Health ID and scores those matches. The data is combined three times, by Medicaid ID, SSN, and Gender/Date of Birth.

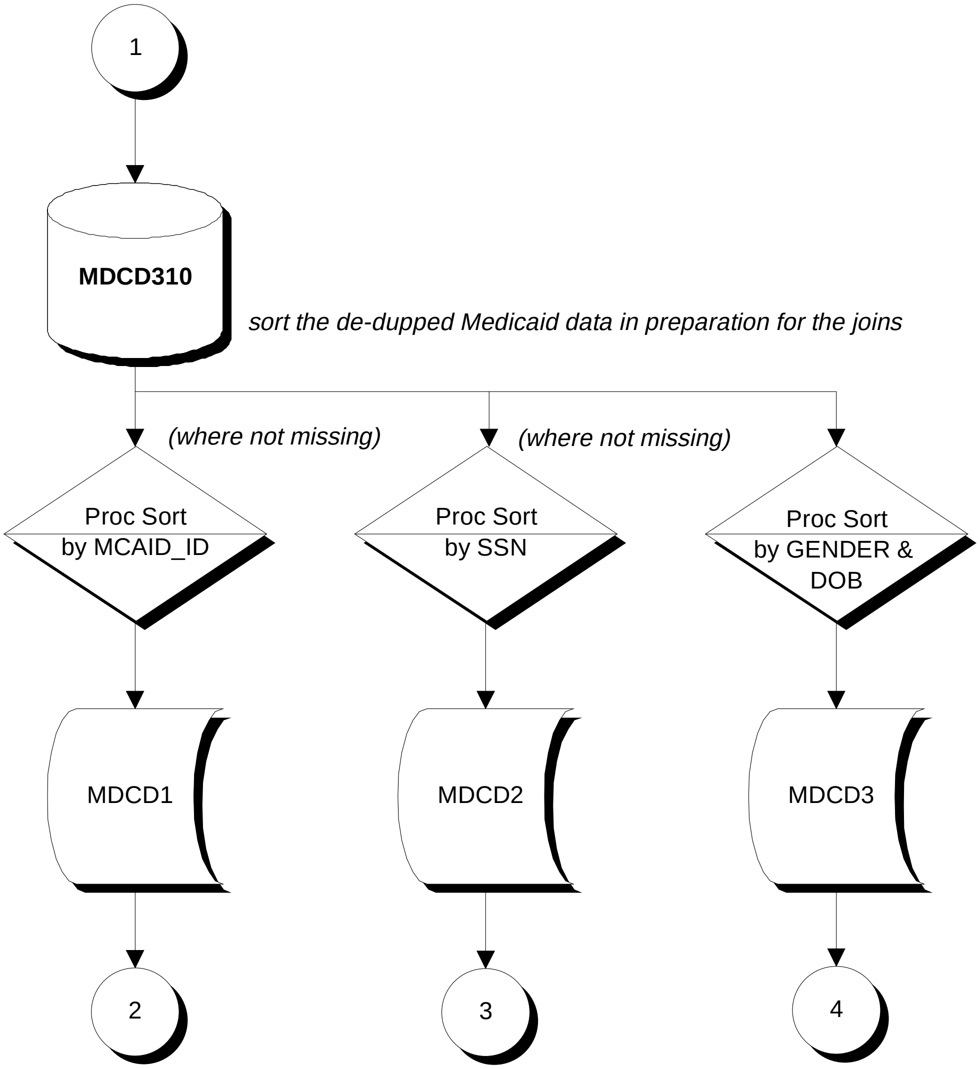


Figure 10 - Program 300 (continued)

Step320: Matches Medicare and MH/AOD Data

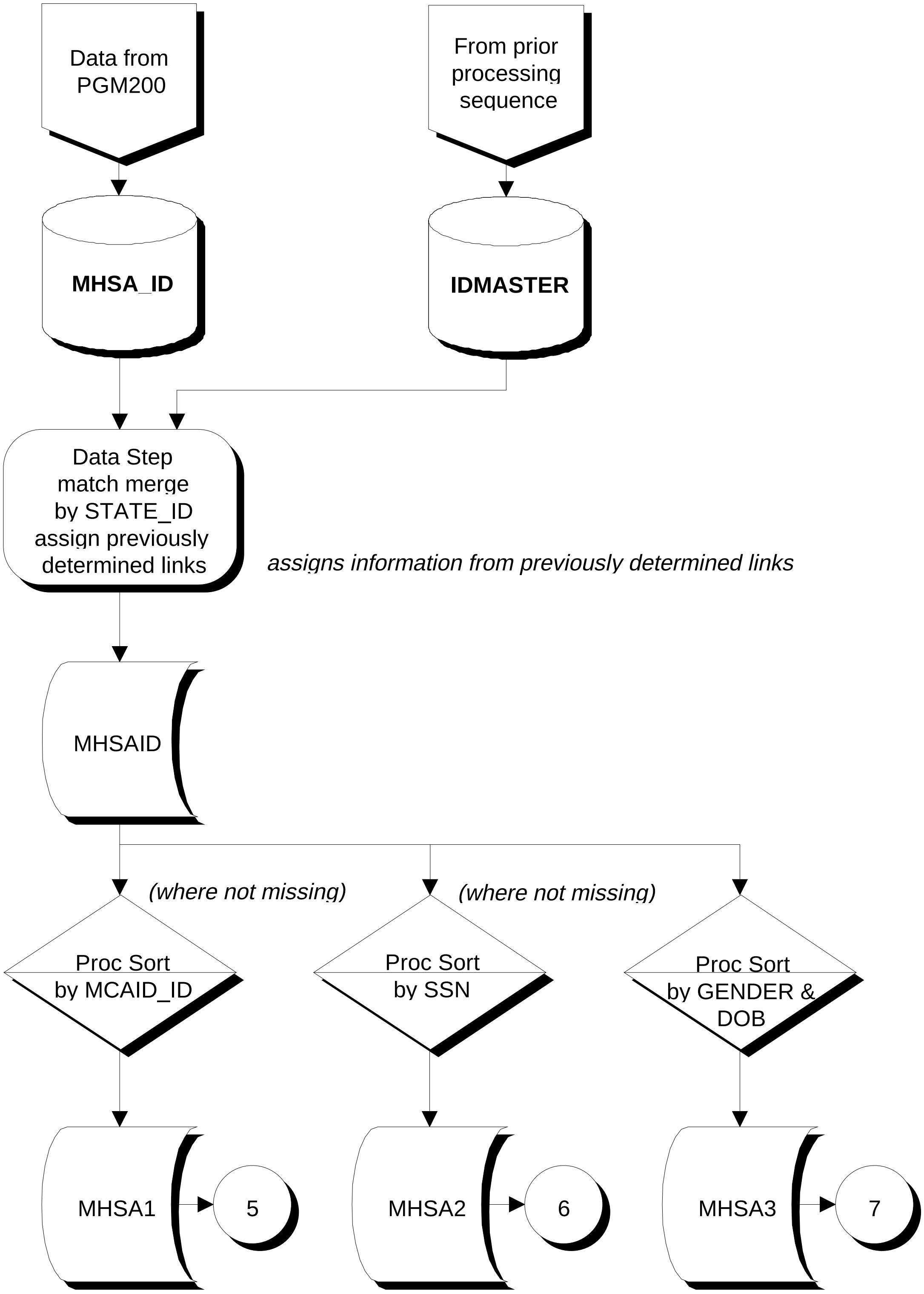

Data from PGM200
From prior processing sequence
MHSA_ID
IDMASTER
Data Step match merge by STATE_ID assign previously determined links
assigns information from previously determined links
MHSAID
(where not missing)
(where not missing)
Proc Sort by MCAID_ID
Proc Sort by SSN
Proc Sort by GENDER & DOB
MHSA1
5
MHSA2
6
MHSA3
7

Figure 10 - Program 300 (continued)

Step320: Matches Medicaid and MH/AOD Data

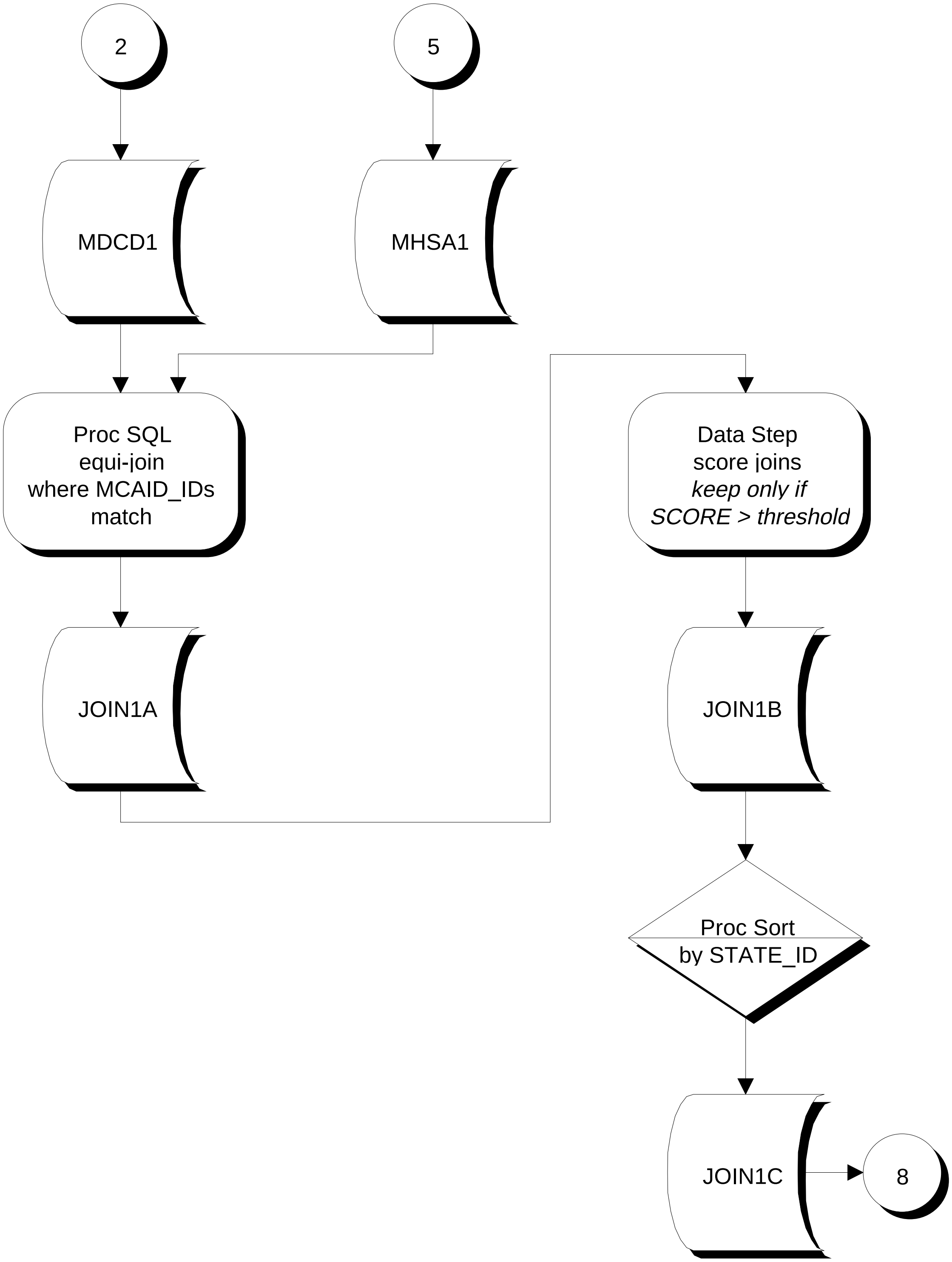


Figure 10 - Program 300 (continued)

Step320: Matches Medicaid and MH/AOD Data

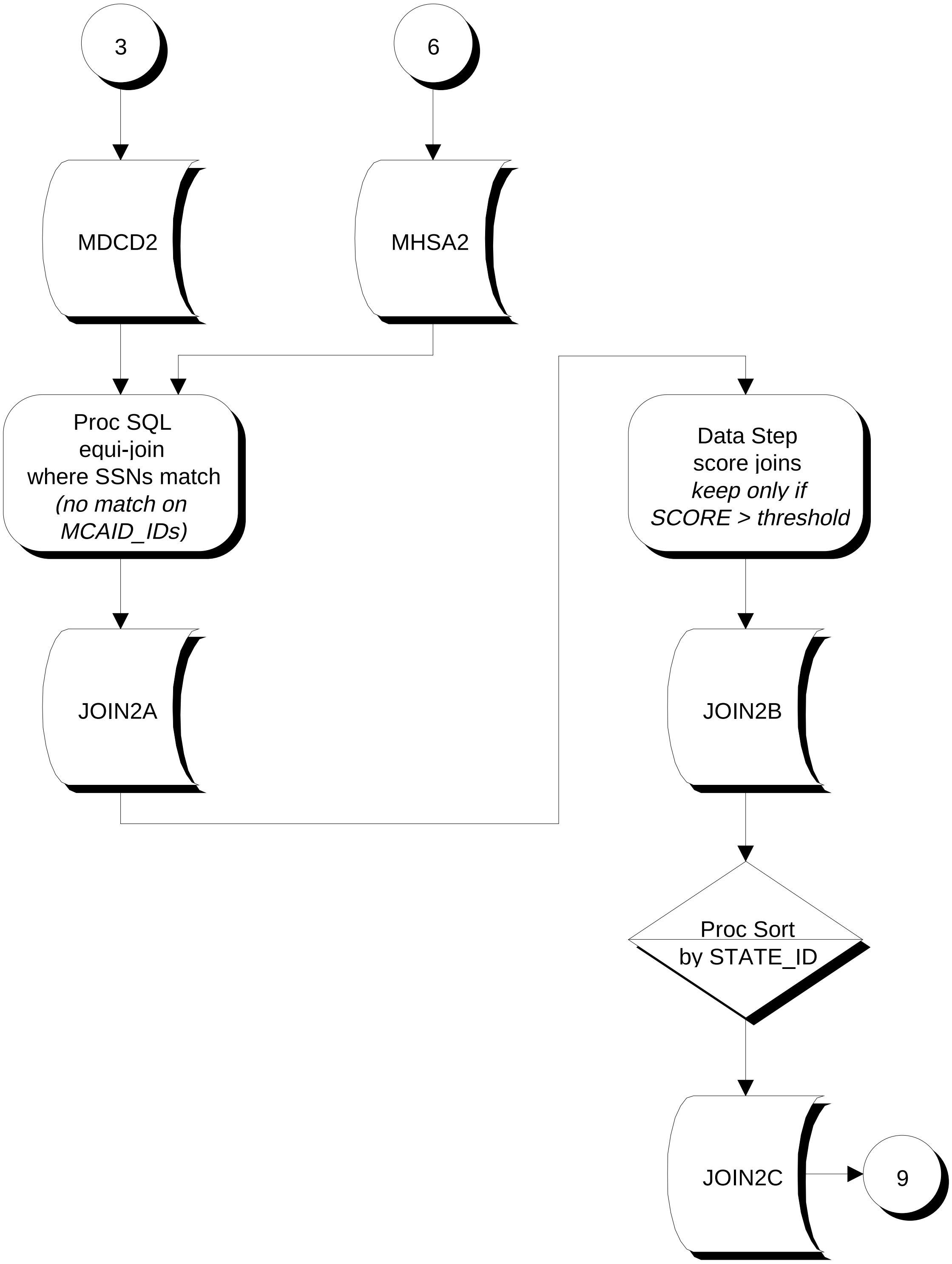


Figure 10 - Program 300 (continued)

Step320: Matches Medicaid and MH/AOD Data

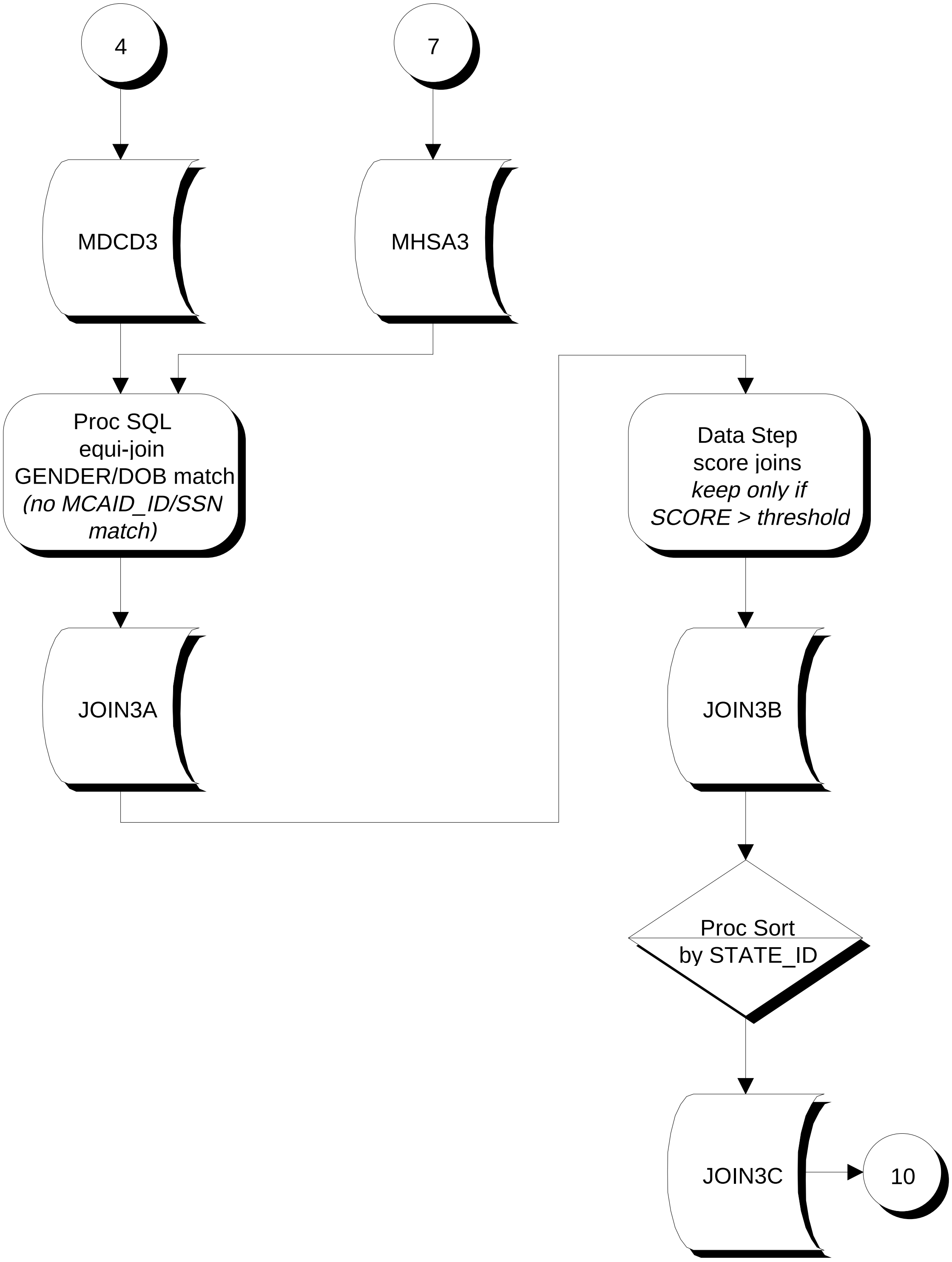


Figure 10 - Program 300 (continued)

Step320: Matches Medicaid and MH/AOD Data

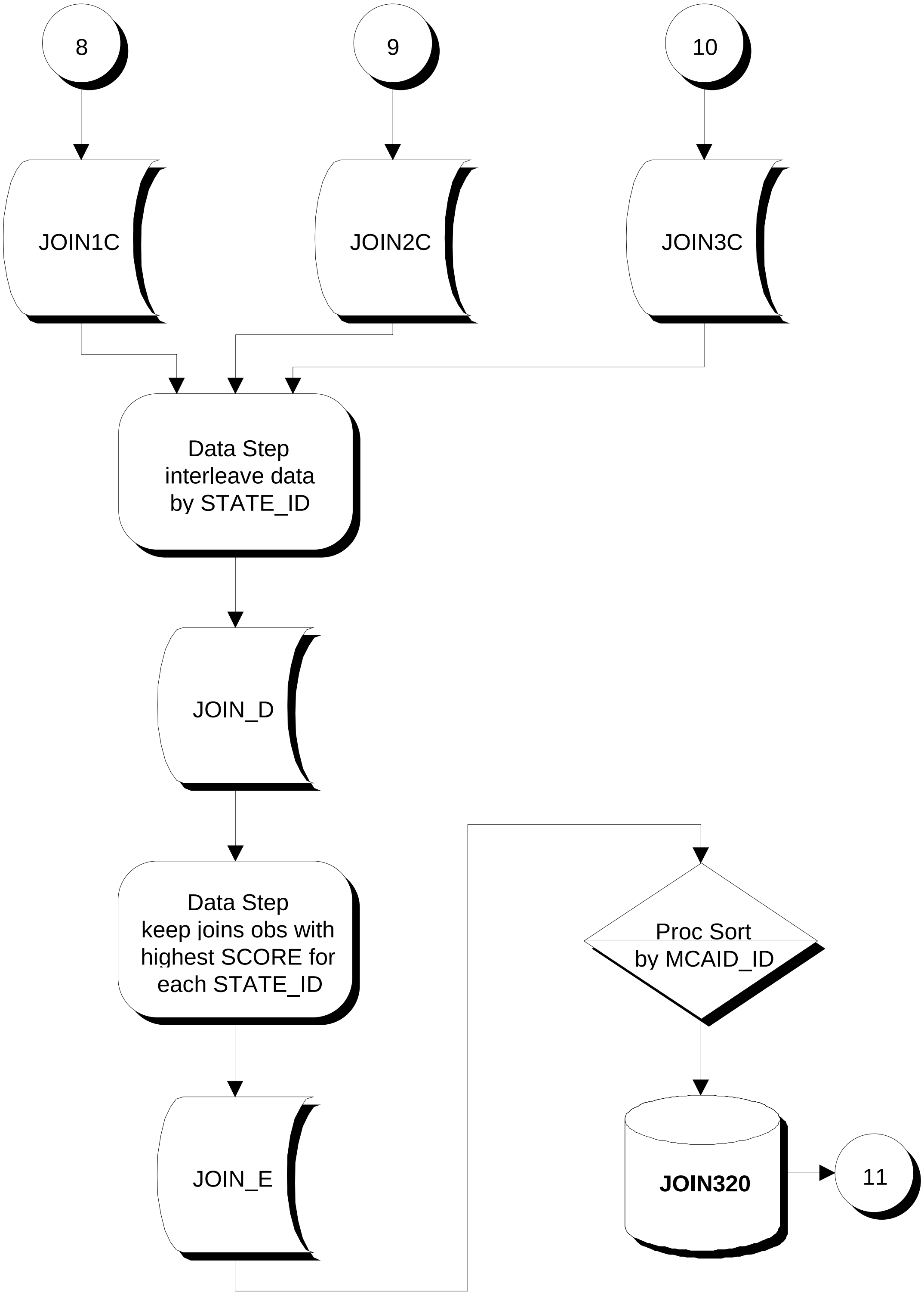


Figure 10 - Program 300 (continued)

Step330: De-duplicate the MH/AOD IDs Not Matching with Medicaid Data

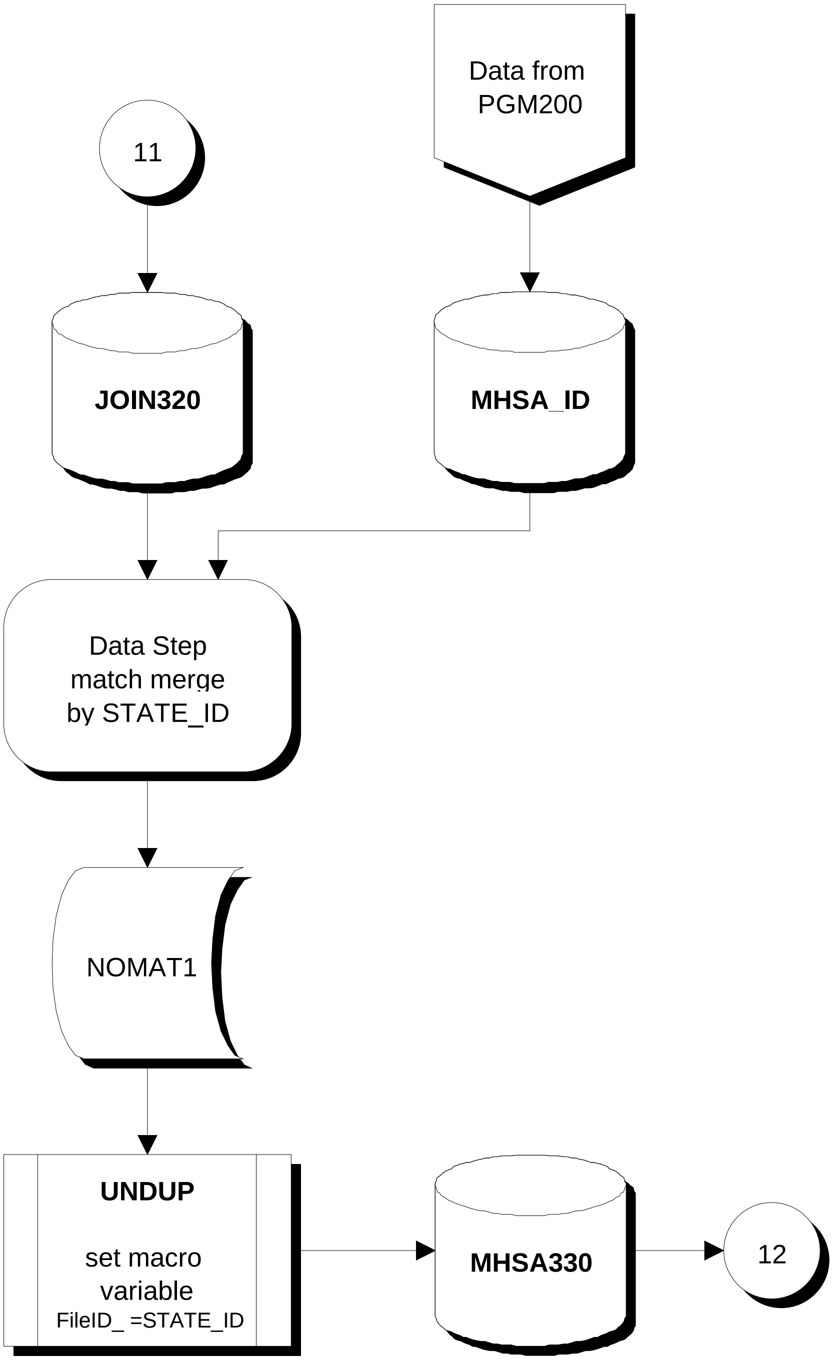
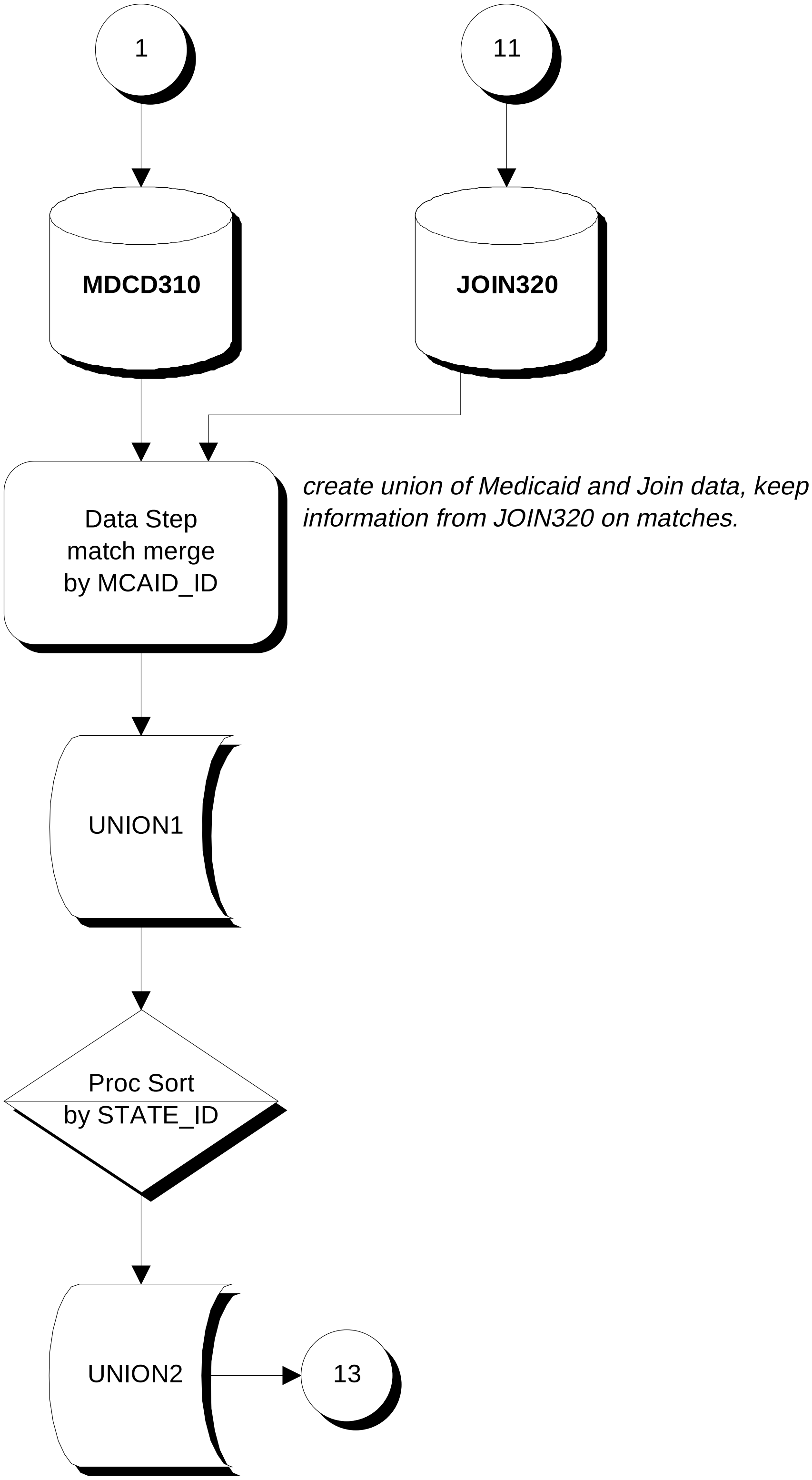


Figure 10 - Program 300 (continued)

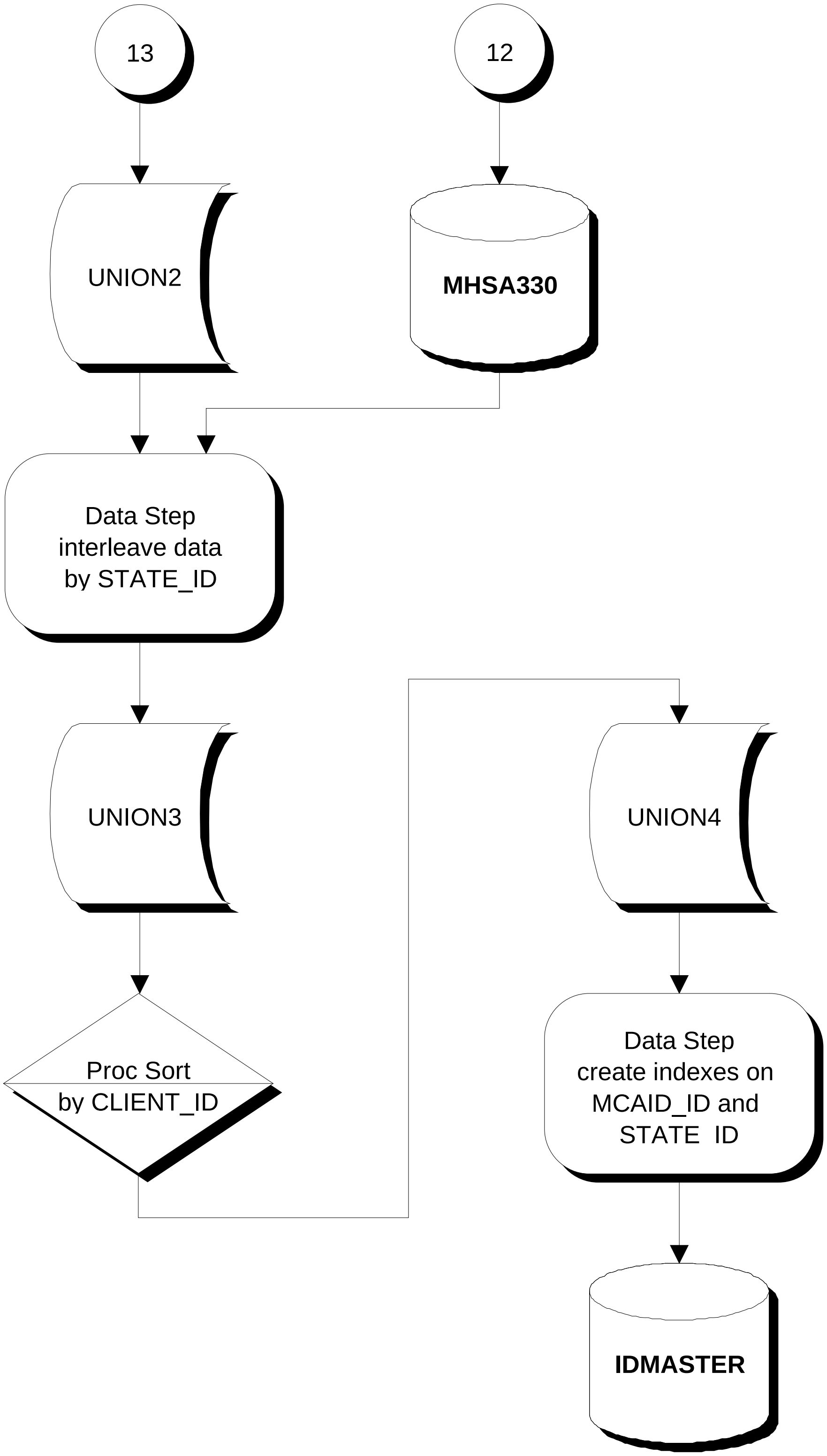
Step340: Combine De-duplicated and Joined ID Data



create union of Medicaid and Join data, keep information from JOIN320 on matches.

Figure 10 - Program 300 (continued)

Step340: Combine De-duplicated and Joined ID Data



UNDUP include

UNDUP will be a SAS include file: code to be *included*, or inserted, within another program. UNDUP will unduplicate clients within a single data set according to the criteria described in the section “The Linking and Unduplication Process”. The code will be generic so that it will work with either the Medicaid ID or MH/AOD ID data sets. In order to make the code generic, global variables will be used to reference

1. the input data set,
2. the output data set, and
3. the file ID variable.

These global variables must be set prior to including UNDUP. By making this code generic, it can be used in both the beginning and end of program 300 to unduplicate first the Medicaid ID data, and later the MH/AOD ID clients that do not link to the Medicaid data. The UNDUP flowchart may be found on page 51.

In addition to unduplicating a data set of IDs, UNDUP also assigns *CLIENT_ID*s. All *CLIENT_ID* assignments will be made sequentially: the first assigned *CLIENT_ID* will be 1, the second assignment will be 2, and so on. The data set, **MAPID**, will track the last ID assigned to guarantee that all *CLIENT_ID*s are unique.

The first task in UNDUP matches all client observations within the input data. Using SQL, the observations will be matched with two or three separate reflexive equi-joins, depending on the file ID. The equi-joins will be made where one of the following criteria are true:

- Medicaid IDs are equal,
- Social Security Numbers are equal, and
- both gender and date of birth are equal.

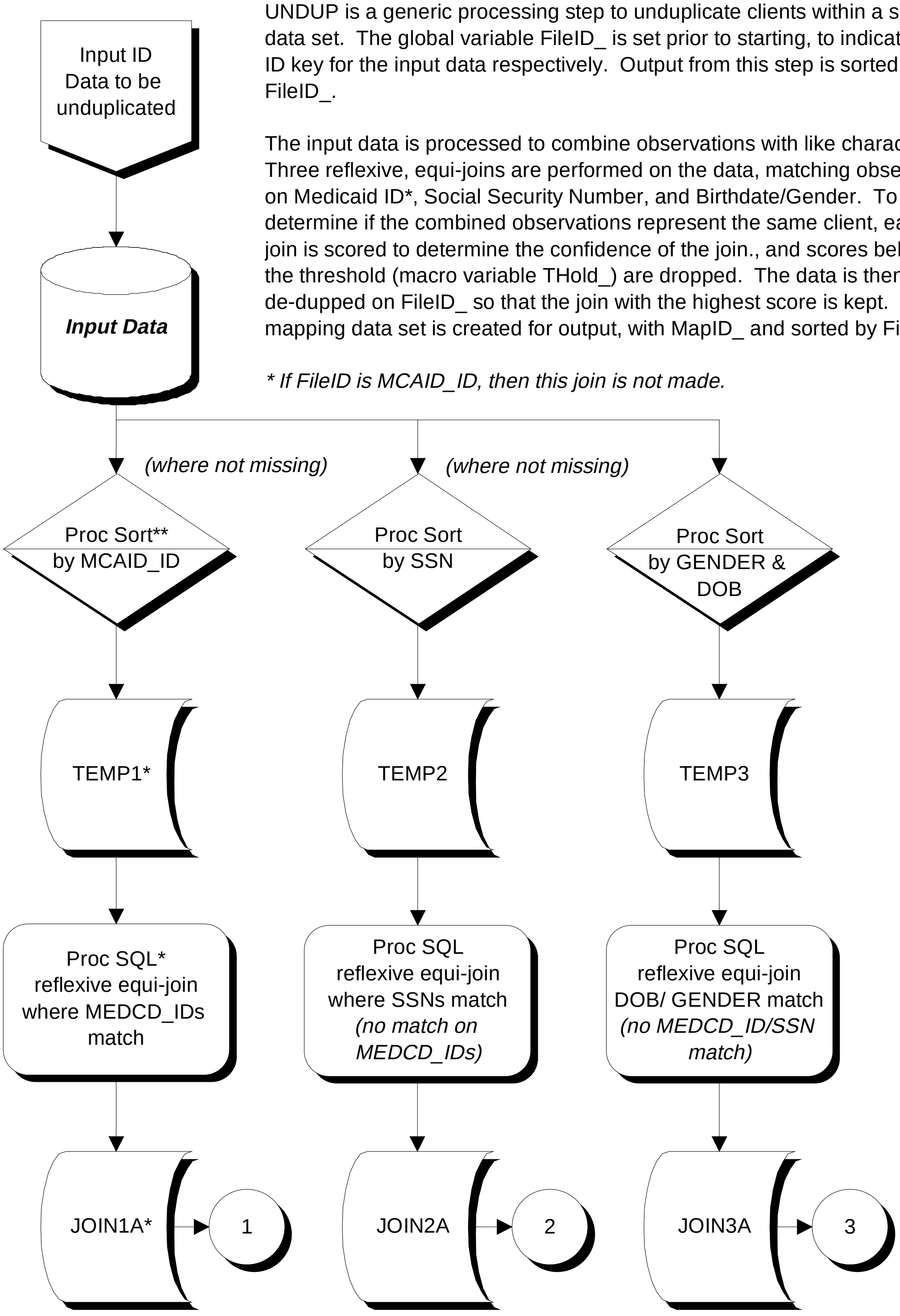
(No join will be made on Medicaid ID when the file ID variable is Medicaid ID). After each join, the matched records will be scored and then sorted by the file ID variable. Scoring is described in the prior section “Scoring Links”. Data from the separate joins will be combined and the match with the highest score within each file ID group will be kept as the mapping for that file ID. Mapped file IDs will be assigned a *CLIENT_ID* at this point if none is present. The last task merges the input data with the unduplicated clients to create the output data. *CLIENT_ID* assignments will be made where there is none present.

Figure 11 - UNDUP Include

UNDUP is a generic processing step to unduplicate clients within a single data set. The global variable FileID_ is set prior to starting, to indicate the ID key for the input data respectively. Output from this step is sorted on FileID_.

The input data is processed to combine observations with like characteristics. Three reflexive, equi-joins are performed on the data, matching observations on Medicaid ID*, Social Security Number, and Birthdate/Gender. To determine if the combined observations represent the same client, each join is scored to determine the confidence of the join., and scores below the threshold (macro variable THold_) are dropped. The data is then de-dupped on FileID_ so that the join with the highest score is kept. A mapping data set is created for output, with MapID_ and sorted by FileID_.

** If FileID is MCAID_ID, then this join is not made.*


Input ID Data to be unduplicated
Input Data
(where not missing)
(where not missing)
Proc Sort** by MCAID_ID
Proc Sort by SSN
Proc Sort by GENDER & DOB
TEMP1*
TEMP2
TEMP3
Proc SQL* reflexive equi-join where MEDCD_IDs match
Proc SQL reflexive equi-join where SSNs match (no match on MEDCD_IDs)
Proc SQL reflexive equi-join DOB/ GENDER match (no MEDCD_ID/SSN match)
JOIN1A*
1
JOIN2A
2
JOIN3A
3

*** If FileID_ is not MCAID_ID*

Figure 11 - UNDUP Include (continued)

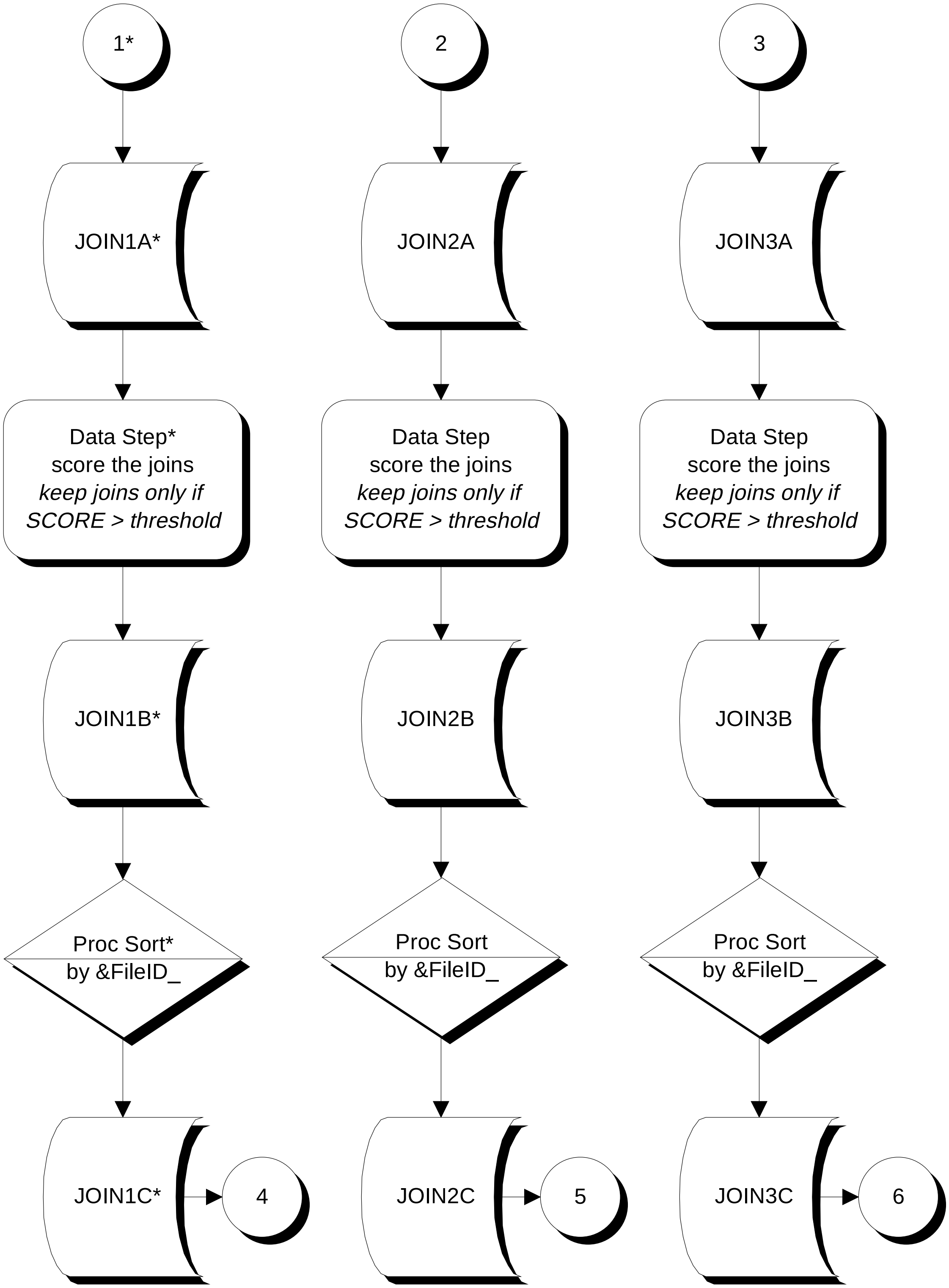


Figure 11 - UNDUP Include (continued)

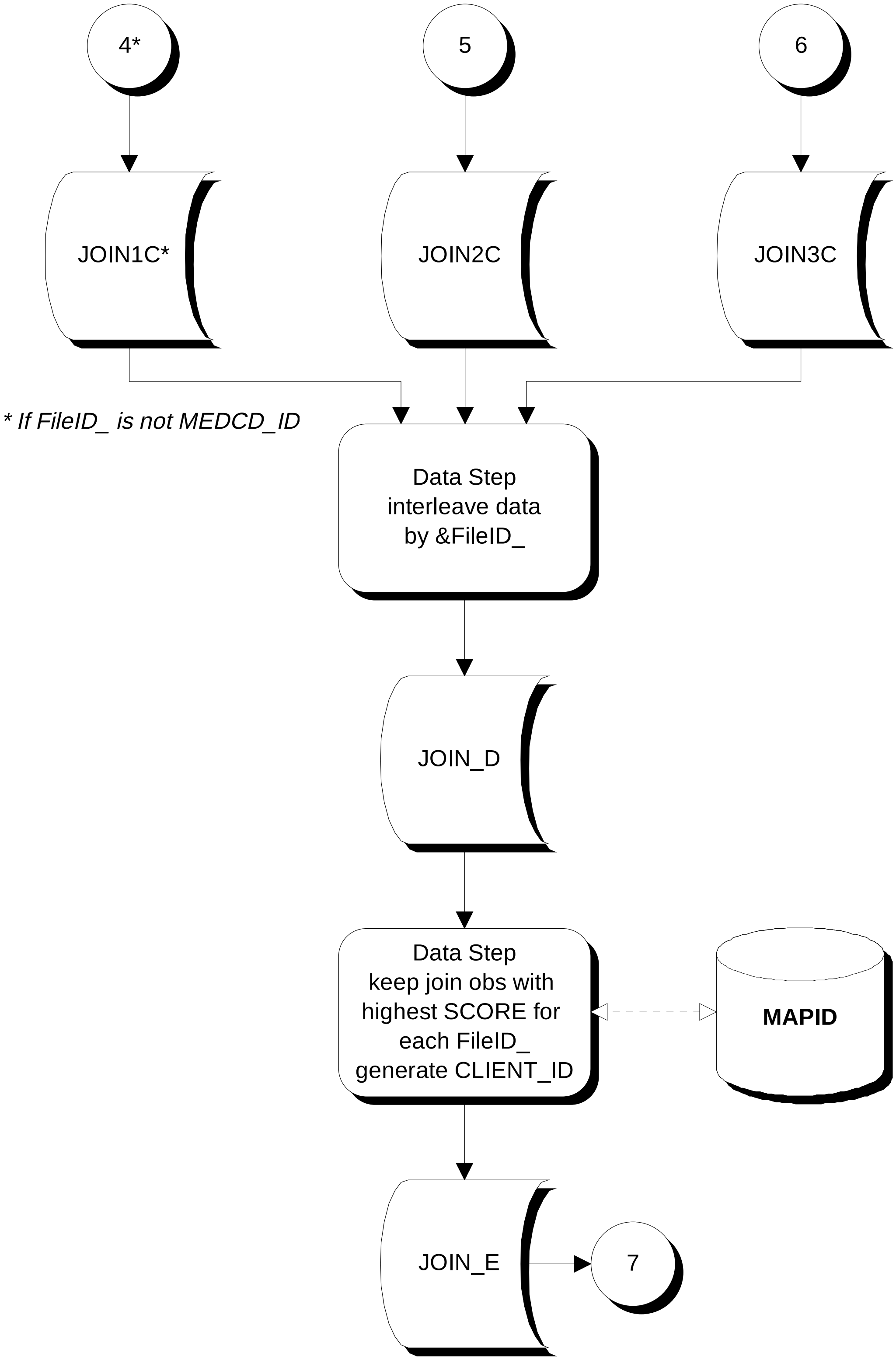
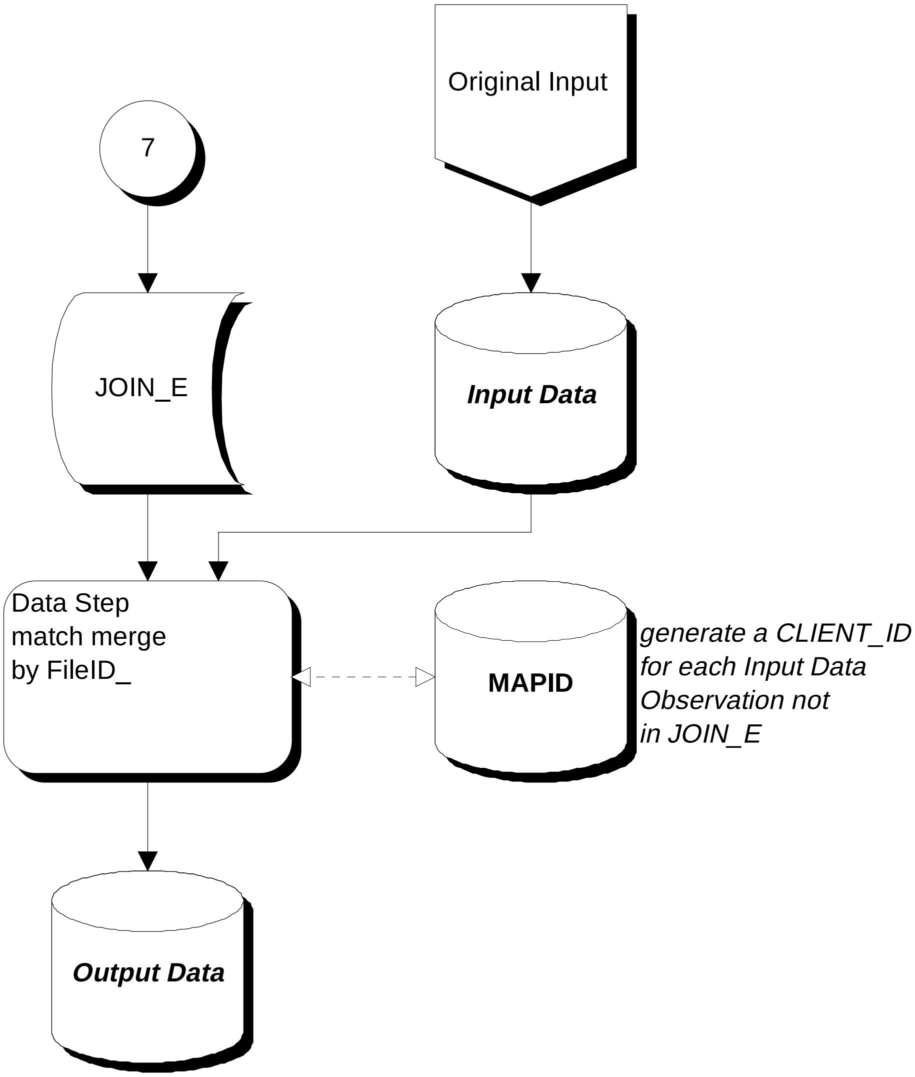


Figure 11 - UNDUP Include (continued)



Program 400

Program 400 will create data sets of Medicaid services and summarized Medicaid activity for clients with MH/AOD services. The services and activity data will cover all medical services for these clients. For inputs, program 400 uses the master ID data set, **IDMASTER**, created in program 300 and two Medicaid data sets from program 100. The two Medicaid data sets are **MLIST**, the list of all recipients of MH/AOD services from the Medicaid claims and encounter files, and **MCLAIMS**, the data set with all Medicaid claims and encounters. Output data sets, the services data **MDCDSERV**, and the summarized activity data **MDCDACTV**, will be sorted by Medicaid ID. If the services data proves too diverse to fit cleanly into a single data set, it will be necessary to divide **MDCDSERV** into several data sets, such as inpatient, outpatient, and prescription drug. Figure 12 beginning on page 56 visualizes the flow of data through program 400.

Processing Steps

Program 400 will collect all Medicaid claims for the MH/AOD population in the final integrated database, regardless of the type or place of the service. Processing for program 400 takes place in two steps. The first step (410) will combine all Medicaid eligible MH/AOD clients - as identified in program 300 - with Medicaid recipients identified with MH/AOD claims in program 100 (**MLIST**), and compile a list of Medicaid IDs for which claim records will be collected. Collection of claims will occur in the second step (420) using the program 100 data set containing all Medicaid claims and encounter data, **MCLAIMS**. This step will resolve adjustments and create stays. A service level data set and a summarized activity data set, both sorted by the Medicaid ID variable (*MCAID_ID*) will be output from program 400.

Figure 12 - Program 400

Step410: Combine Medicaid Eligible MH/AOD Clients (from PGM300) with Medicaid who have MH/AOD Claims (from PGM100)

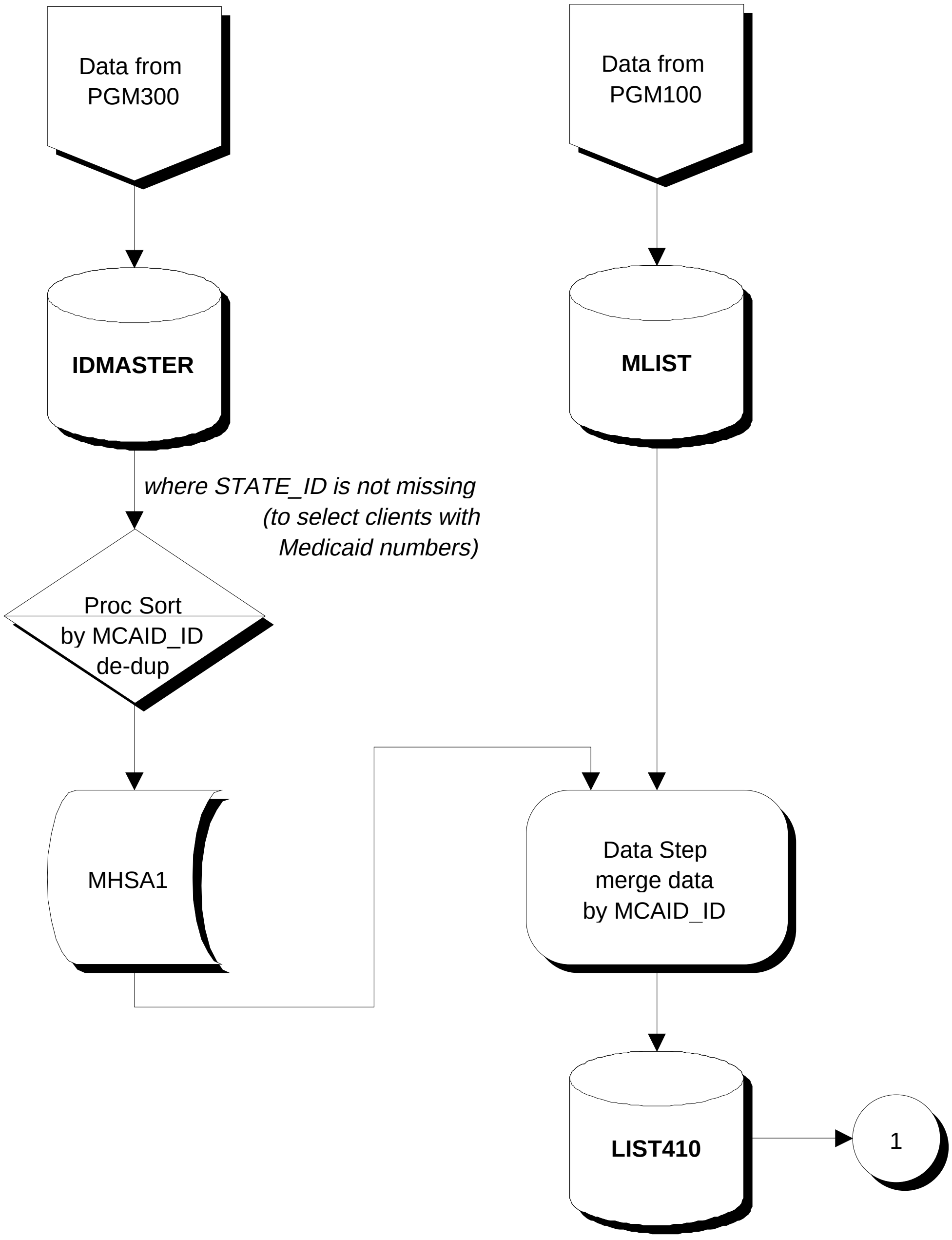
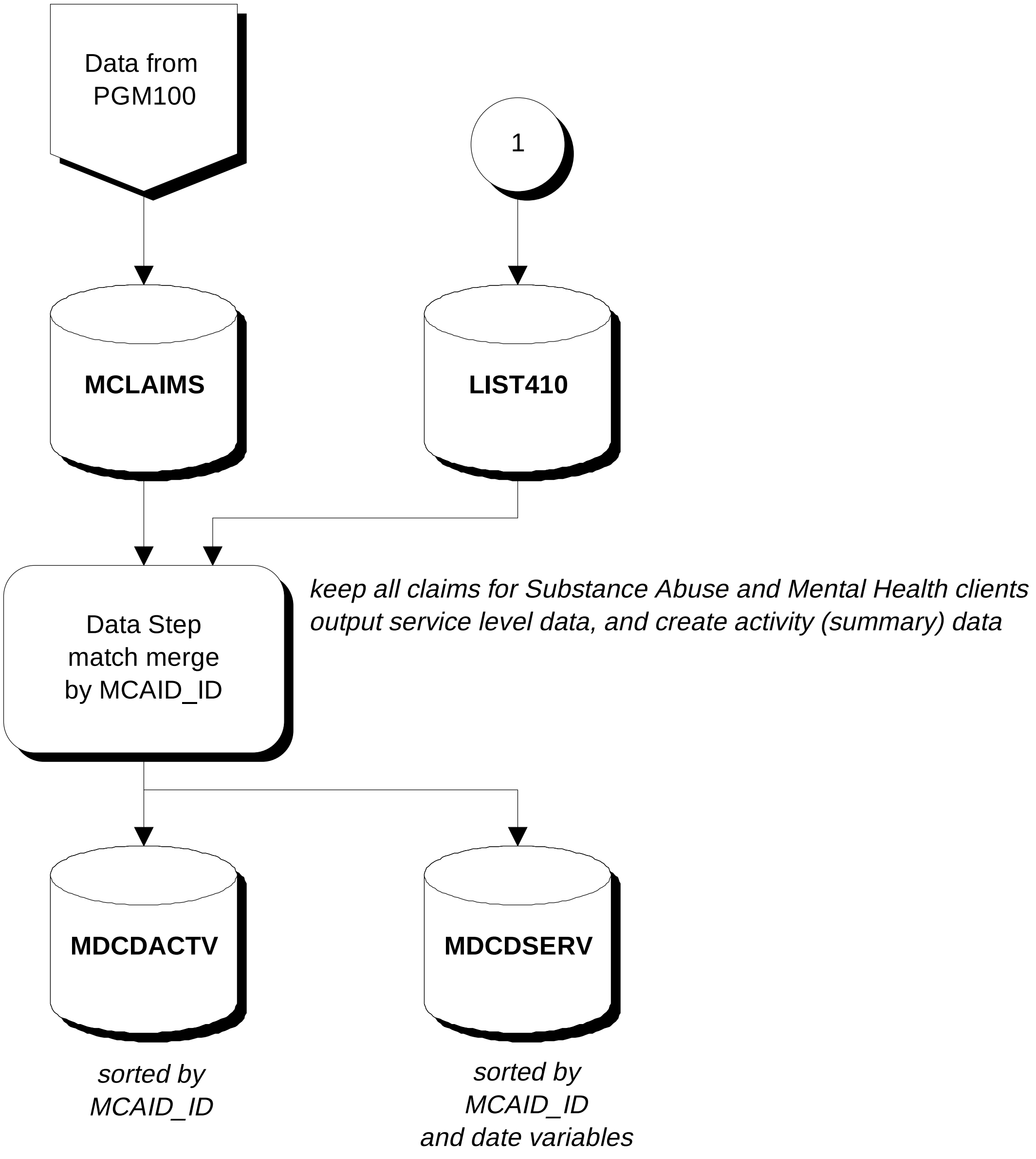

Data from PGM300
Data from PGM100
IDMASTER
MLIST
where STATE_ID is not missing (to select clients with Medicaid numbers)
Proc Sort by MCAID_ID de-dup
MHSA1
Data Step merge data by MCAID_ID
LIST410
1

Figure 12 - Program 400 (continued)

Step420: Create Claims and Activity Files For Identified Clients

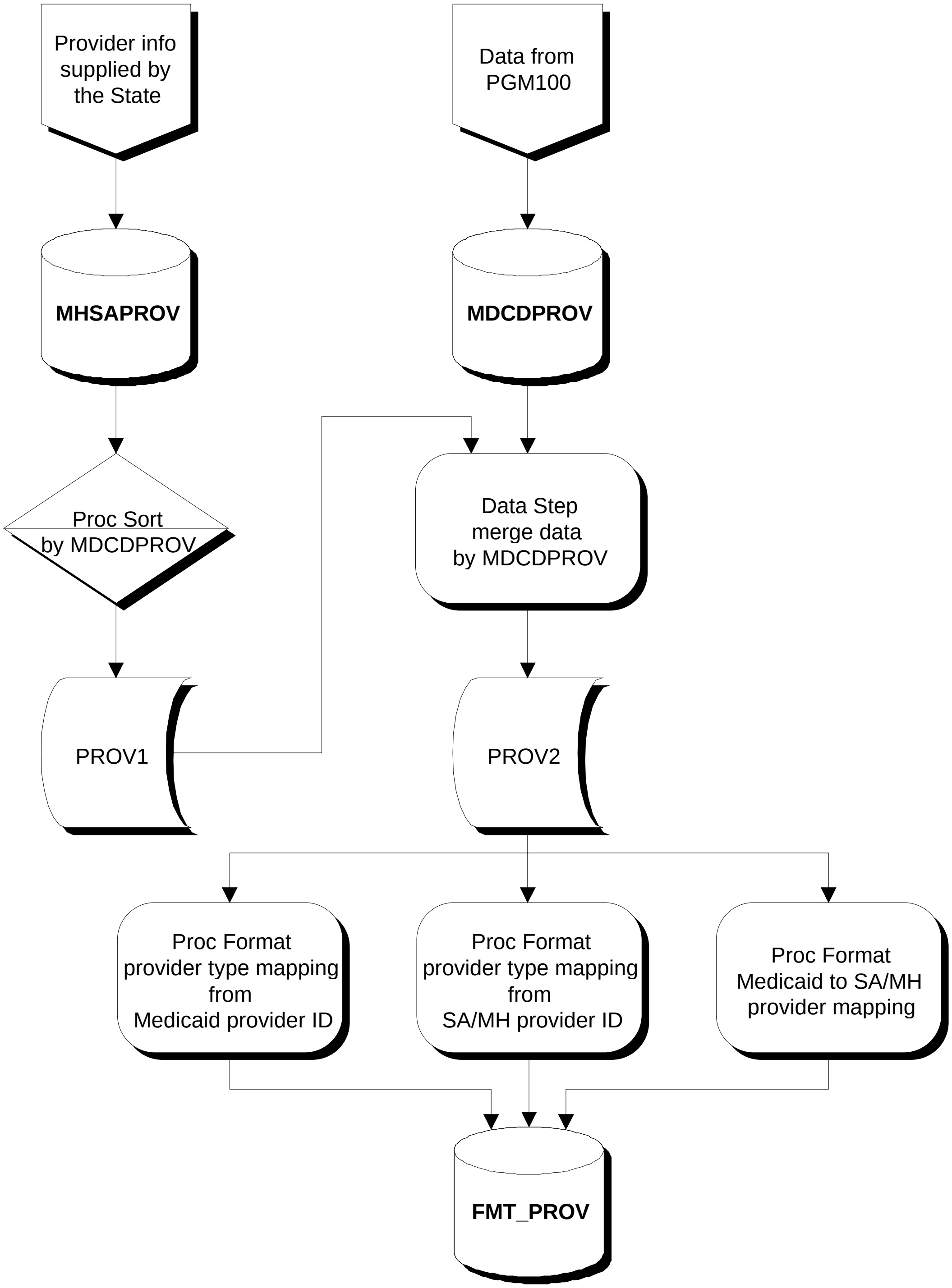


Program 500

Program 500 is a single step program which will process provider information and create SAS formats for use creating the integrated database in program 600. **`MDCDPROV`**, the Medicaid provider data SAS loaded in program 100, will be combined with MH/AOD provider information, and SAS formats mapping from Medicaid to MH/AOD number will be created. The formats will be saved in the SAS format library **`FMT_PROV`** and used in program 600 to unduplicate those services counted in both the Medicaid and MH/AOD data. The flowchart for program 500 is shown in figure 13 on page 59.

Figure 13 - Program 500

Step510: Add MH/AOD Provider Information to the Medicaid Provider Information and Create


Provider info supplied by the State
Data from PGM100
MHSAPROV
MDCDPROV
Proc Sort by MDCDPROV
Data Step merge data by MDCDPROV
PROV1
PROV2
Proc Format provider type mapping from Medicaid provider ID
Proc Format provider type mapping from SA/MH provider ID
Proc Format Medicaid to SA/MH provider mapping
FMT_PROV

Program 600

Program 600 will combine data sets from earlier programs and create the integrated database for each of the three participating states. The integrated database will consist of three data sets: **`SERVICES`**, **`ACTIVITY`**, and **`CLIENTS`**. The first of these data sets, the `SERVICES` data set, will contain service information, MH/AOD and otherwise, where that data is available. The second data set, **`ACTIVITY`**, will summarize all MH/AOD and other medical services to the client and month[6] level. Both the **`SERVICES`** and **`ACTIVITY`** data sets will contain multiple observations for each *`CLIENT_ID`*. **`CLIENTS`**, the third data set will contain demographics data with one observation for each *`CLIENT_ID`*. All three databases will be sorted by *`CLIENT_ID`*.

In creating the integrated database, program 600 will use formats created in program 500 and data from programs 200, 300, and 400. The services and summarized activity data sets of MH/AOD data from program 200 (**`MHSASERV`** and **`MHSAACTV`**) and of Medicaid data from program 400 (**`MDCDSERV`** and **`MDCDACTV`**) will be combined into the integrated database's **`SERVICES`** and **`ACTIVITY`** data sets. SAS formats from program 500 will be used to help unduplicate services found in both the Medicaid and MH/AOD data. From the "master" ID data set, **`IDMASTER`**, created in program 300, the **`CLIENTS`** data set will be created.

Unduplication of Services and Service Counts

A single service can appear on both the MH/AOD and Medicaid data because some services provided by the state MH/AOD agency are partially funded by that state's Medicaid program. Services recorded by Medicaid but administered by the MH/AOD agency present record duplication problems. The Medicaid claims or encounter data will contain records of these services, and MH/AOD data will also record the usage. If the Medicaid and MH/AOD data were simply combined, services and summarized counts would be incorrect. Duplication would exist within the data, and usage would be overcounted. It is during program 600 that corrections will be made for duplicate services to prevent such a situation. Along with the linking of client records, this is the most challenging operation of the project, and one not attempted to this extent by any other group or project. The feasibility of this methodology will be better assessed after we receive data from the states.

[6] The degree of summarization will depend upon the data received from the participating states. If it is not possible to summarize to monthly activity, then another measure, such as quarter or year, will be used.

Where the MH/AOD agency provides claim or service level data, Unduplication will occur based on provider numbers, dates, and procedure codes. The SAS formats created in program 500 will be used to map Medicaid provider numbers to MH/AOD provider numbers. In these instances, it is likely that the Medicaid and MH/AOD claims will each contain information which is missing on the other. When unduplicating claims, information from both sources will be combined to provide as much claim information as possible. A detailed decision tree will be used to determine which source to use when both sources provide the same type of information, but different values. For example cost information from a Medicaid claim will be used in place of inferred costs from MH/AOD data.

In those instances where only summarized data is available from the MH/AOD agency, formats mapping Medicaid provider numbers to type of provider will be used to determine if the Medicaid claim contains information that is also in summarized MH/AOD data, corrections will be made accordingly. Information, such as costs, obtained from the Medicaid claim will be used to update the summarized activity data set

Processing Steps

The first step (610) of program 600 builds the **`SERVICES`** data set. Combining the MH/AOD and Medicaid services data, **`MHSASERV`** and **`MDCDSERV`**, and using the provider ID formats from program 500, Step 610 will unduplicate services, keeping all detail information available between the two sources. Duplicate service information will be retained for use in Unduplicating the summarized activity data sets in the next step (620). Step 620 will combine the summarized activity data sets (**`MHSAACTV`** and **`MDCDACTV`**) of MH/AOD and Medicaid data and build the **`ACTIVITY`** data set of the integrated database. This data will be unduplicated using the list of duplicated services from step 610 and the generalized Unduplicating rules mentioned in the preceding section. The **`CLIENTS`** data set will be built in the final step (630), using **`IDMASTER`** and the **`ACTIVITY`** data set built in the prior step to build an ID data set. **`CLIENTS`** will contain one observation for each client in the **`ACTIVITY`** data set, as well as demographic information. No identifiers, such as names or Social Security numbers, will appear on the **`CLIENTS`** data set, nor will identifiable data such as date of birth appear.

Figure 14 - Program 600

STEP610: Combine the MH/AOD and Medicaid Service Data, Creating the Services Data Fi

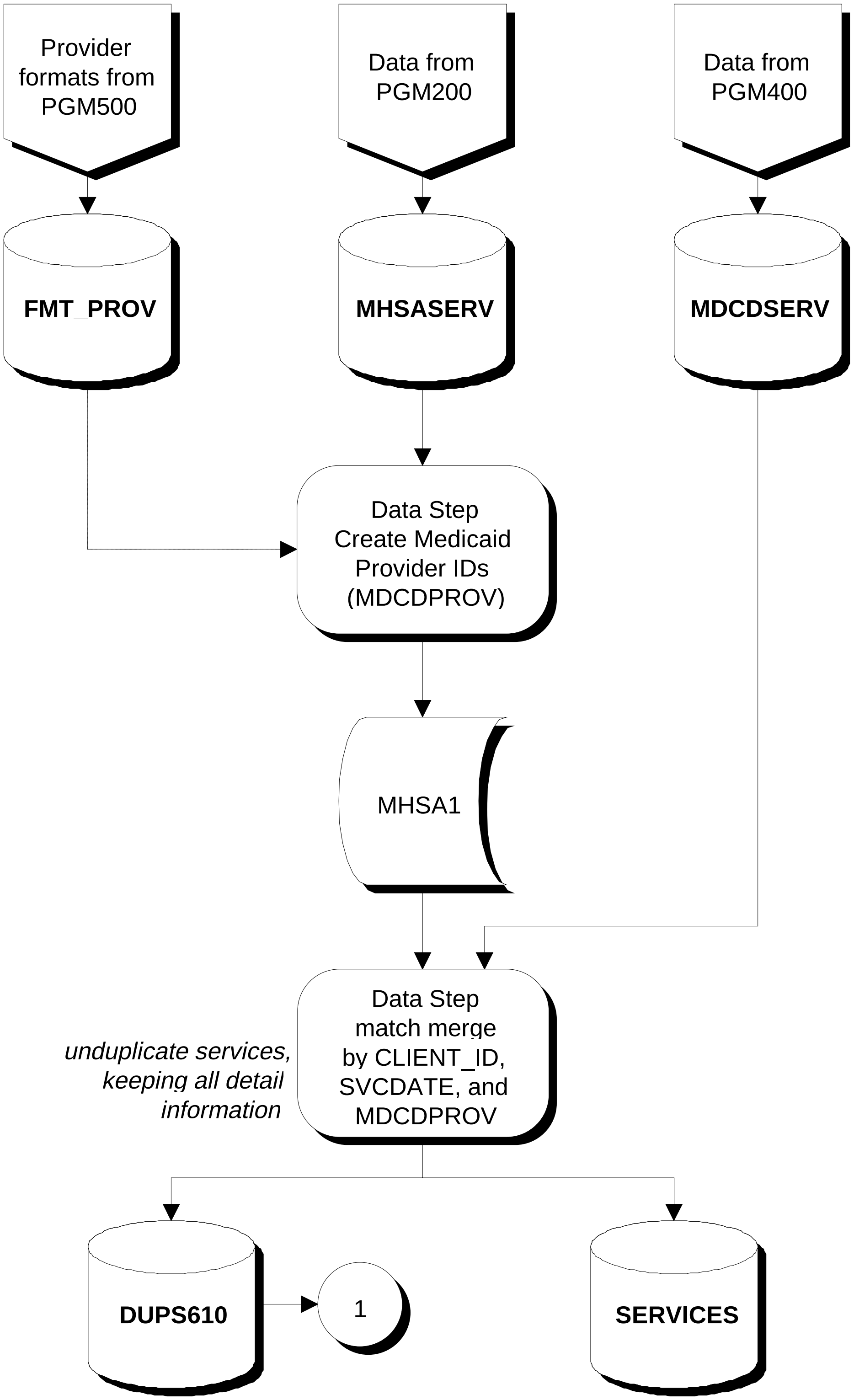

Provider formats from PGM500
Data from PGM200
Data from PGM400
FMT_PROV
MHSASERV
MDCDSERV
Data Step Create Medicaid Provider IDs (MDCDPROV)
MHSA1
unduplicate services, keeping all detail information
Data Step match merge by CLIENT_ID, SVCDATE, and MDCDPROV
DUPS610
1
SERVICES

Figure 14 - Program 600 (continued)

STEP620: Combine the MH/AOD and Medicaid Activity Data. Creating the Activity Data F

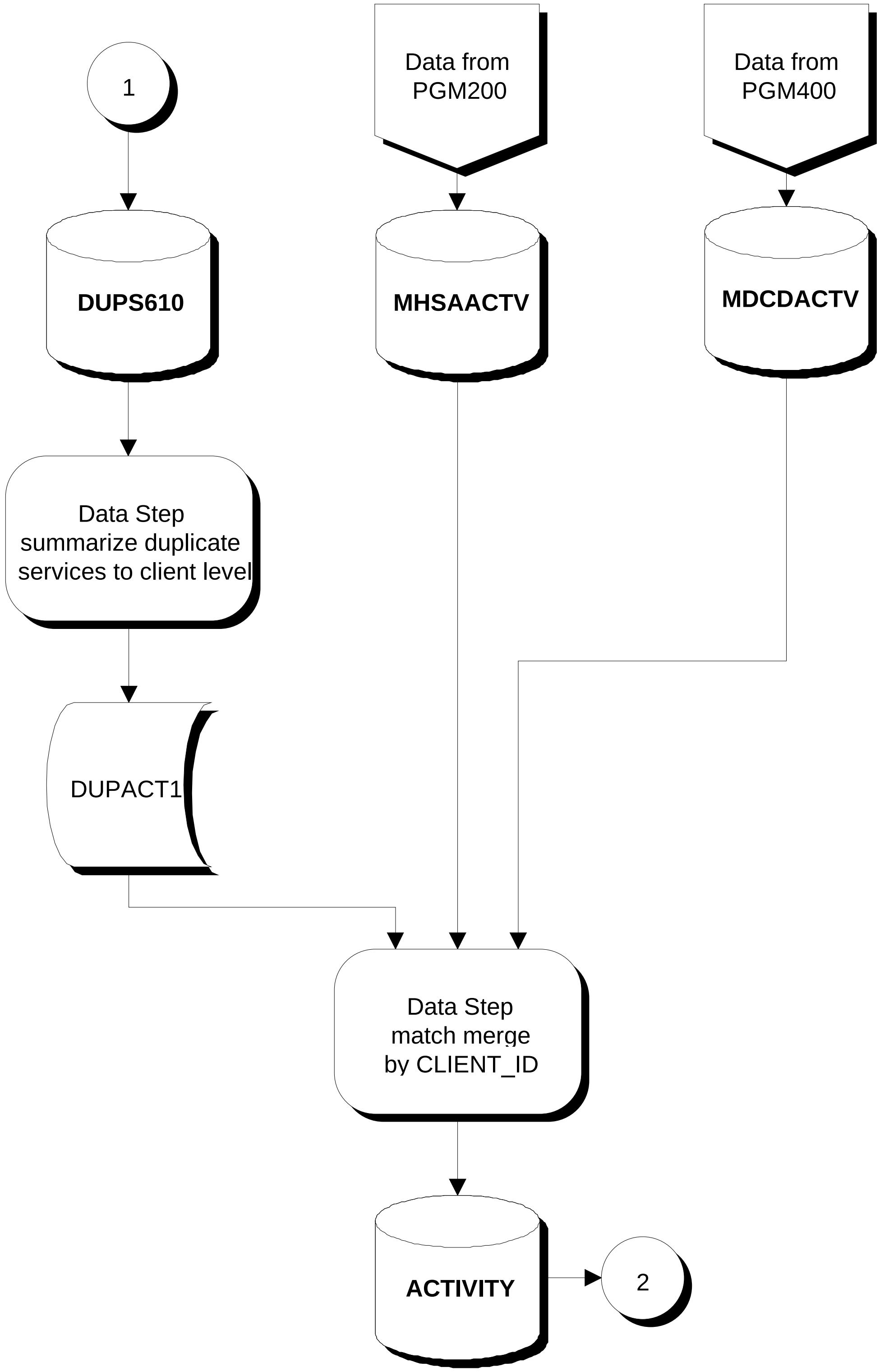
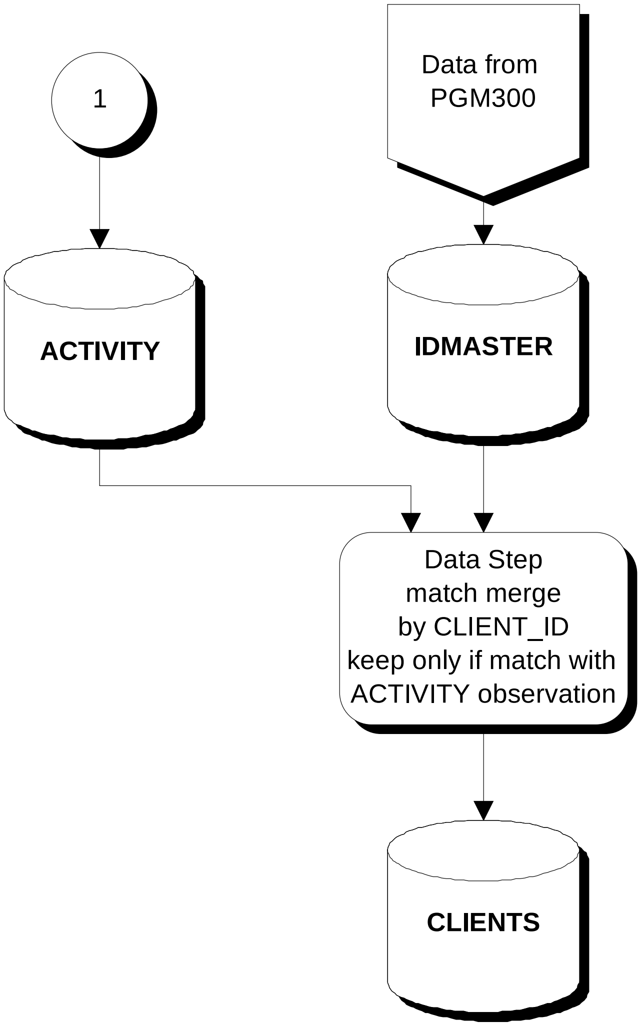

1
Data from PGM200
Data from PGM400
DUPS610
MHSAACTV
MDCDACTV
Data Step summarize duplicate services to client level
DUPACT1
Data Step match merge by CLIENT_ID
ACTIVITY
2

Figure 14 - Program 600 (continued)

STEP630: Create the Client Data File



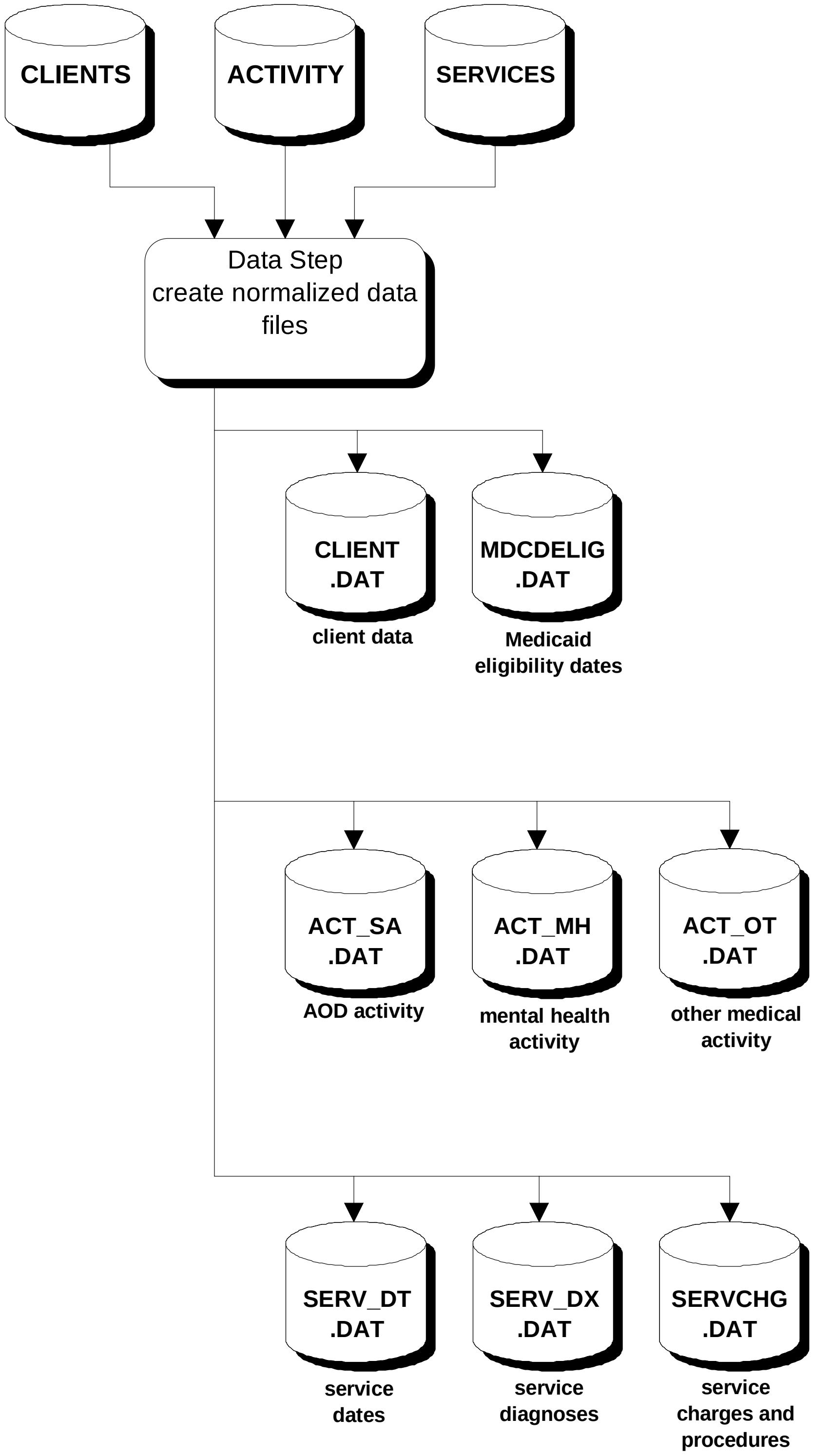
Program 700

Program 700 will perform the postproduction tasks of creating normalized, ASCII text files from the integrated database. These files will be available for states who plan to use the integrated database within a relational database management system (RDBMS). Input to program 700 will be the three data sets that comprise the integrated database: **`CLIENTS`**, **`ACTIVITY`**, and **`SERVICES`**. Output from the program will be a series of third normal form text files, ready for importing to virtually any RDBMS or analysis software, containing:

- client demographic information from the **`CLIENTS`** data set,
- Medicaid eligibility information from the **`CLIENTS`** data set,
- summarized mental health activity data from the **`ACTIVITY`** data set,
- summarized AOD activity data from the **`ACTIVITY`** data set,
- summarized activity for other medical services from the **`ACTIVITY`** data set,
- service diagnoses from the **`SERVICES`** data set, and
- service procedures from the **`SERVICES`** data set.

Figure 15 on page 66 shows an example of the types of files envisioned. The actual number and content of files is dependent on the data received from the participating states, and some changes to the above list are likely.

Figure 15 - Program 700



Database Reporting

The SAS System has been chosen as the architecture of the integrated database. In addition to being a database management system, SAS is a full-featured analysis and presentation application. It contains a variety of statistical and data display procedures, many of which will be used in building the integrated database. These SAS procedures will be used for two purposes: quality assurance and documentation - tasks well suited to SAS. Quality assurance reports, using intermediate data, will be used during the building of the database to monitor progress. Additional reports will be generated after completing the integrated database. Frequency distributions for categorical variables and selected univariate statistics (such as ranges, mean, and median values) for continuous variables will be generated and included in the database documentation. Figures 16 and 17 show two examples of possible database reports[7].

Figure 16 - Sample Demographic Report

```
                        Total Clients    MH Services      AOD Services
                        -------------    -----------      ------------
Total                   652,336          533,937          363,901

White                   350,613  53.7%   301,414  56.5%   138,283  38.0%
Black                   141,242  21.7%   121,272  22.7%   100,254  27.5%
Hispanic                 59,904   9.2%    35,930   6.7%    43,008  11.8%
Native American          44,814   6.9%    29,400   5.5%    36,264  10.0%
Asian/Pacific Islander   33,883   5.2%    27,473   5.1%    26,155   7.2%
Unknown Race             21,880   3.4%    18,448   3.5%    19,937   5.5%

Male                    324,189  49.7%   207,140  38.8%   195,851  53.8%
Female                  319,030  48.9%   321,349  60.2%   164,381  45.2%
Unknown Gender            9,117   1.4%     5,448   1.0%     3,669   1.0%

Age 0-17                 99,138  15.2%    90,211  16.9%    42,225  11.6%
Age 18-64               434,367  66.6%   356,155  66.7%   241,895  66.5%
Age > 64                101,677  15.6%    79,071  14.8%    71,127  19.5%
Unknown Age              40,196   2.6%     8,500   1.6%     8,654   2.4%
```

[7] Numbers in the sample reports are purely fictional.

Figure 17 - Sample Client Services Matrix

```
Clients with:
------------------------------------------------------------------------
                  | Medicaid Services | MH Services       | AOD Services
------------------+-------------------+-------------------+-------------
Medicaid Services | N/A               | 501,937           | 349,865
                  |                   |    94.0%          |    96.1%
------------------+-------------------+-------------------+-------------
MH Services       | 501,937           | N/A               | 293,003
                  |    76.9%          |                   |    80.5%
------------------+-------------------+-------------------+-------------
AOD Services      | 349,865           | 293,003           | N/A
                  |    53.6%          |    54.9%          |
```

The integrated database will be structured to facilitate analysis and reporting. The majority of this work will take place at SAMHSA, and at agencies of the participating states, and we anticipate that researchers and policy makers will use the database for a variety of statistical tasks and ad hoc reporting. To facilitate such use, indexes will be created on select variables within the SAS database, and sample programs will be provided in the documentation. The sample programs will demonstrate straightforward, common tasks, providing analysts with a head start towards many database uses. The indexes, meanwhile, will improve the speed and efficiency of most queries and analyses.

Anticipated Resource Requirements

Due to the volume of records anticipated, based on the data inventory of Task 21, and the number of data sets planned in constructing the integrated databases, we estimate that storage for input data, the final integrated databases, and intermediary data will require 20 GB (gigabytes) of disk space for each state. Additional work space, of nearly 30 GB, will be needed for processing in order to accommodate the numerous sorts and joins. An equally robust processor, or CPU, is also called for. In building each database, hundreds of millions of observations will be processed, a volume of data demanding a powerful and efficient processor. We therefore will use the DEC AlphaServer running DEC UNIX and SAS, as detailed in the Task 23 report. The DEC has both the storage and the processing power for this project.

The project will process data from three states and eight different agencies. Each state agency will provide multiple data files. A significant amount of time will be needed for programming, testing, and revisions. We estimate that 120 days of full time programming will be required to build, test, and review the individual SAS "steps" or programs described above. Testing will be performed with small data samples to validate the execution of this process. An additional 60 days will be needed to construct the prototypes: assembling the perl scripts, or "programs", testing the entire process, and creating the prototype databases using the test data. Programming will most likely be tasked to three or four programmers. The allocation of programmer's time will be relatively heavy during the early stages. This is because it is easier, and less costly, to correct errors at the beginning of a process than later on. Through extensive testing during the initial stages, we expect to minimize the corrections and changes needed later in the project.

Conclusion

In this report, we have detailed the system requirements, and proposed processing design, for building integrated databases of MH/AOD services for three states: Delaware, Oklahoma, and Washington. An overview of the design described the processes necessary to construct the databases. Expected issues and difficulties were considered, and three crucial operations were identified. Those three operations are:

- associating costs with MH/AOD services,
- linking client records, and
- unduplicating Medicaid and MH/AOD services.

We also gave a brief examination of database documentation, including our inclination towards creating an HTML based intranet for the documentation system. The overview offered a high-level view in order to provide a complete picture of the process. With the overall process described, we proceeded on to the particulars of each processing stage. Program by program, we examined details of the process. Each discussion presented the flow of data and addressed the crucial operations described above. A thorough examination of the client linking process, including the scoring criteria, was presented. And finally, we discussed resources needed by the project, in both computer hardware and programmer time.

Much effort has been spent to provide a well thought out and detailed processing plan. While this design is as complete as possible at this stage of the project, it is only a starting point. The design is not “carved in stone.” Revisions may prove necessary once we receive sample data and begin testing. Areas where revisions are likely include:

- client linking methodology,
- criteria for scoring client links, and
- unduplicating services and service counts.

Detailed design and comprehensive testing during the initial stages of building the databases are crucial to this project. Consequently, we are placing our emphasis on planning and design. This will lead to a shortened development time by minimizing the corrections and changes needed later in the project, and also increase the quality of the final integrated databases.

Appendix A - Draft Criteria[8] for Identifying MH/AOD Conditions

ICD-9-CM diagnosis codes	Mental Health (including Alzheimer's) 290, 293-302, 306-316, 331.0
	Substance Abuse (and related medical conditions) 265.2, 291, 292, 303-305, 357.5, 357.6, 425.5, 535.3, 571.0-571.3, 648.3, 648.4, 655.4, 655.5, 760.7, 779.5, 962.0, 965.0, 967, 968, 969, 977.0, 977.3, 980
	Potentially alcohol-related conditions -- need to rule out other conditions (pellagra, cerebral degeneration, epilepsy, cirrhosis/liver disease without mention of alcohol.) 265.0, 331.7, 334.4, 345.0-345.9, 571.4-571.9, 780.1, 780.3, V154
	Mentally retarded 317-319
ICD-9-CM procedure codes	94.1-94.6
CPT-4 procedure codes	83840, 90801, 90841-90847, 90849, 90853, 90855, 90857, 90862, 90870, 90871, 90899
UB92 revenue codes	114, 124, 134, 144, 154, 116, 126, 136, 146, 156, 204, 513, 900-904, 909, 910-917, 919, 944, 945, 961
Provide and Service Types	State specific codes to identify the following service providers: • Psychiatrists • Psychologists • Mental Health Clinics • Substance Abuse/Detoxification Clinics • Psychiatric Hospitals • Substance Abuse Hospitals • Psychiatric Acute Care, but not a hospital • Mental Institutions

[8] A condensed draft of the proposed criteria, developed as part of SAMHSA's Medicare, Medicaid, and Managed Care Analysis Project (Contract 280-95-0011).

Appendix B - Glossary

C2 Security - A standard from the US Government National Computer Security Council (an arm of the U.S. National Security Agency), "Trusted Computer System Evaluation Criteria, DOD standard 5200.28-STD, December 1985" which defines criteria for trusted computer products.

Denormalize – To allow redundancy in a table to that a database table can remain flat, rather than normalized. Data is often denormalized for statistical analysis and OLAP.

Equi-join – A type of join where table rows are combined only if the value of a column in the first table is equal to the value of the column in the second table. In a SQL expression, a WHERE clause or and ON clause contain the column names that must satisfy the requirement.

ER diagrams (**E**ntity-**R**elation diagrams) – A type of diagram used in data modeling for relational databases.

HTML (**H**yper**T**ext **M**arkup **L**anguage) - the publishing language of the World Wide Web. HTML is a simple to use document markup language that is completely platform independent.

Join – Combining data from two or more tables, resulting in a single, new table or view. A general join combines each row from one table with all rows of the other tables, forming the Cartesian product of the original tables.

Logical design – The phase of a database design concerned with identifying the relationships among the data elements.

Normalize – The process of removing redundancy by separating data into multiple tables. One of the constraints imposed on a relational database.

OLAP (**O**n-**L**ine **A**nalytic **P**rocessing) – A loosely defined set of principles that provide a framework for decision support. Generally using a multidimensional database model with appropriate access and analysis tools, OLAP primarily involves aggregating large amounts of diverse data. OLAP can involve millions of data items with complex relationships. Its objective is to analyze these relationships, look for patterns, trends, exceptions, and turn them into meaningful information. An OLAP system should support logical and statistical processing of results without the end user having to submit the request in language like SQL or C++.

OLTP (**O**n-**L**ine **T**ransaction **P**rocessing) – The process of capturing transactions, and the data relevant to them, in real time. An OLTP is primarily concerned with adding, updating, inserting and deleting records. Most frequently used with reference to relational databases.

One-to-many relationship – A logical data relationship in which the value of one data element in a given table can exist in combination with many values of a data element in another table, but not vice versa.

perl (**P**ractical **E**xtraction and **R**eport **L**anguage) - is a general purpose, interpreted scripting language available on UNIX systems, useful for executing system calls and other commands.

Physical design – The phase of database design following the logical design that identified the actual database tables and index structures used to implement the logical design.

Query – A request for information, generally as a formal SQL command passed from the front-end application to a database or search engine.

RDBMS (**R**elational **D**ata **B**ase **M**anagement **S**ystem) – A database that organizes and accesses data according to relationships between data items. Based on the relational model developed by E.F. Codd, RDBMS define data structures, storage and retrieval operations, and integrity constraints. The data and relations are organized in tables, which are collections of records. Each record in a table contains the same fields. Records in different tables may be linked if they have the same value in one particular field in each table.

Reflexive join – Joining a single table with itself.

RISC (**R**educed **I**nstruction **S**et **C**omputer) – A processor whose design is based on the rapid execution of a sequence of simple instructions rather than on the provision of a large variety of complex instructions (as in a Complex Instruction Set Computer).

SQL (**S**tructured **Q**uery **L**anguage) – A language which provides a user interface to an RDBMS. SQL is the de facto standard for RDBMS and can be embedded in other programming languages. SQL also provides basic functions for defining and manipulating tables of data.